FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

Data Mining em aplicações de Desenho racional de Fármacos

Eduardo José Valadar Martins

VERSÃO DE FINAL

Mestrado Integrado em Engenharia Informática e Computação

Orientador: Rui Camacho

Co-Orientador: Nuno Fonseca

25 de Fevereiro de 2017

Data Mining em aplicações de Desenho racional de Fármacos

Eduardo José Valadar Martins

Mestrado Integrado em Engenharia Informática e Computação

25 de Fevereiro de 2017

Resumo

O desenvolvimento de um fármaco novo é um processo longo, muito caro e levanta algumas questões éticas.

A criação de um novo medicamento passa por quatro fases: desenvolvimento de uma nova molécula, testes pré-clínicos, testes clínicos e aprovação regulatória. O uso de técnicas de *Data Mining* (DM) na fase de desenvolvimento pré-clínico pode acelerar e reduzir os custos da criação de um novo fármaco. Esta fase está dividida em três sub-fases: farmacologia, ordenação toxicológica e formulação, para as quais propomos a utilização de técnicas de *Data Mining* para otimizar o processo.

Algoritmos preditivos (classificação e regressão) são aplicados em dados disponíveis em repositórios de compostos já testados nas fases farmacológicas e de ordenação toxicológica, com base em testes ADMET (Absorção, Distribuição, Metabolismo, Excreção, Toxicidade). As regras definidas como úteis, por parte de um especialista, podem ser úteis para o processo de desenho de novos medicamentos. Desta forma técnicas de previsão DM podem usar resultados conhecidos de testes anteriores e fazer previsão sobre as propriedades toxicológicas e de ADME.

Em suma, pretende-se desenvolver uma interface entre um especialista e as ferramentas de *Data Mining*, que permita uma interação com os repositórios de moléculas, construir modelos com recurso ao DM, visualizar os modelos num formato usado pelos bioquímicos e permitir interação para modificar os modelos construídos.

Abstract

The development of a new drug is a long process, very expensive and raises some ethical questions.

The creation of a new medicine goes through four phases:drug discovery, pre-clinical development, clinical development and regulatory approval.The use of Data Mining (DM) techniques in pre-clinical development phase can accelarate and cheapen the creation of a new drug.This phase is divided in three subphases: pharmacology, toxicological ordination and formulation, in which will be used the DM techniques to optimize the process.

The classification and regression algorithms, as well as the techniques of a data mining to use will use repositories compouds already tested by pharmacological phases and taxicological ordination, in based of ADMET tests, and the rules defined as useful, by one skilled, to the design process of new drugs. In that case the techniques of data mining prediction can use known results of previous tests and make a prediction about the taxicological properties and ADME.

In short, it is intended to develop an interface between an expert and data mining tools that allow interaction interface with the repositories of molecules, build models using the Data Mining, view the models in a format used by biochemists and allow interaction to modify the built models.

Agradecimentos

Em primeiro lugar, gostaria de agradecer ao orientador da tese Professor Rui Camacho do Departamento de Engenharia Informática da Faculdade de Engenharia da Universidade do Porto. Ele consistentemente permitiu que esta tese fosse o meu próprio trabalho, orientando-me na direção certa sempre que achou que eu precisava.

Finalmente, devo expressar minha profunda gratidão aos meus pais, à minha namorada Mariana e ao meu colega Rúben Cordeiro por me fornecerem apoio infalível e encorajamento contínuo ao longo destes anos de estudo. Não esquecendo os meus amigos Alexandre Nascimento, João Correia e José Amaral que sempre me acompanharam e ajudaram nestes últimos dois anos. Esta realização não teria sido possível sem o contributo de todos eles. Obrigado.

Eduardo Martins

Conteúdo

Lista de Figuras

LISTA DE FIGURAS

Lista de Tabelas

Abreviaturas e Símbolos

ADMET	Absorção Distribuição Metabolismo Excreção Toxicidade
WEKA	Waikato Environment for Knowledge Analysis
SMILES	simplified molecular-input line-entry system
SDF	structure-data file
ARFF	Attribute-Relation File Format
K-NN	k-nearest neighbors
SVM	Máquina de vetores de suporte
ROC	Receiver operating characteristic
API	Application programming interface
JSON	JavaScript Object Notation
XML	Extensible Markup Language
MCV	Model-Controller-View
HTTP	Hypertext Transfer Protocol
GUI	Graphical user interface

Capítulo 1

Introdução

O processo de desenvolvimento de um novo fármaco consiste em desenvolver uma molécula que interaja com uma molécula biológica responsável por uma doença de forma a aumentar, diminuir ou inibir a sua função e desencadear assim um efeito terapêutico.

Este processo divide-se em quatro etapas: **descoberta do fármaco**, **desenvolvimento pré-clínico**, **desenvolvimento clínico** e **aprovação regulatória**. Estas etapas geram muita informação que na maior parte das vezes não é aproveitada, mas com recurso a técnicas e algoritmos de *Data Mining* (DM), esses dados podem ajudar a construir modelos úteis para o processo de desenho de novos fármacos e desta forma otimizar os custos e tempo necessários.

Estes algoritmos vão ser essencialmente usados na etapa **Desenvolvimento Pré-Clínico** onde é recebido como *input* um conjunto de moléculas da etapa anterior. Pretende-se que cada molécula do conjunto seja otimizada de forma a aumentar a sua seletividade, aumentar o seu grau de atividade e garantir o nível de atividade pretendido no organismo. Para garantir o nível de atividade pretendido, as moléculas serão avaliadas por testes `ADMET`[1], que visam avaliar as características influentes na exposição do fármaco ao organismo e identificar os compostos tóxicos.

Os testes `ADMET` são feitos através de métodos *in vitro* ou em animais, onde são executadas várias iterações até se alcançar um resultado pretendido. O que para além de levantar questões éticas, também é moroso e dispendioso.

Para mitigar este problema, pretende-se utilizar bases de dados de compostos já conhecidos e estudados, cujos resultados em testes ADMET são conhecidos, e aplicar a esses dados algoritmos de *Machine Learning* (ML), ajudem a explicar e prever os resultados dos testes `ADMET` e desta forma inferir sobre a sua atividade no organismo. Desta forma novas moléculas podem reduzir/evitar os testes em animais.

Um dos objetivos do trabalho realizado é facilitar a interação entre o especialista bioquímico e as ferramentas de manipulação, visualização e desenho de novos fármacos, de forma a não ser necessária a intervenção de especialistas informáticos. Esta ferramenta interliga diferentes

[1] Absorption, Distribution, Metabolism, Excretion and Toxicity

funcionalidades, que normalmente são tratadas independentemente, de forma a torná-la robusta e completa.

1.1 Motivação e Objetivos

A etapa de Desenvolvimento Pré-Clínico é demorada devido, em grande parte, aos testes **ADMET** necessários geralmente realizados em animais. Sendo assim, a utilização de técnicas e algoritmos de *Data Mining* que prevejam os resultados diminuem o tempo necessário a despender em testes e por conseguinte os custo inerentes neste processo. Uma vez que as ferramentas existentes requerem, em problemas mais complexos, a intervenção de um especialista informático, pretende-se facilitar a acessibilidade às funcionalidades de manipulação, visualização e geração de fármacos por parte de um especialista bioquímico.

Este projeto pretende interligar diferentes ferramentas utilizadas na etapa de **Desenvolvimento Pré-Clínico** de um fármaco e tornar a sua utilização transparente por parte de um especialista bioquímico.

Pretende-se desenvolver uma aplicação que recorra a algoritmos e técnicas de *Data Mining* para construção de modelos preditivos de forma a diminuir o tempo gasto em testes laboratoriais. Pretende-se também que o especialista tenha à sua disposição ferramentas de manipulação e visualização, onde poderá realizar alterações à molécula do modelo e reavaliar o efeito das suas alterações.

A aplicação será capaz de interpretar diferentes formatos de entrada, convertendo-os se necessário, calcular descritores de moleculares, construir modelos preditivos, manipular e visualizar os modelos calculados.

1.2 Estrutura da Dissertação

Para além da Introdução, esta dissertação contém mais 4 capítulos.

O **Capítulo 2** introduz os conceitos principais do domínio de desenho relacional de fármacos usando técnicas de *Data Mining*. Nele são analisados os tipos de ficheiros usados para armazenar informação de conjuntos de moléculas, ferramentas de cálculo de descritores moleculares e para executar técnicas de extração de conhecimento, assim como a descrição de todo o processo de extração de conhecimento de um conjunto de dados.

No **Capitulo 3**, é feita uma análise da aplicação desenvolvida. São identificados os componentes da aplicação e as suas funcionalidades em detalhe, seguido da identificação das etapas de extração de conhecimento presentes na aplicação.

No **Capítulo 4**, são apresentadas de forma detalhada as interfaces que compõem a aplicação, assim como as transições entre elas e as interações que o Utilizador pode realizar.

O **Capítulo 5** contém as conclusões do trabalho realizado e enumera um conjunto de possíveis melhorias ao trabalho realizado.

Capítulo 2

Quimio-informática e Data Mining

2.1 O Processo de Desenho Racional de Fármacos

Formatos de Ficheiros Moleculares

Os conjuntos de moléculas a usar nas ferramentas, podem aparecer em ficheiros com diferentes formatações e com diferentes extensões [Wik17a].

Diferentes formatos estão associados a diferentes usos do conjunto de moléculas, podendo efetuar-se conversões entre estes através de ferramentas de conversão de formatos moleculares [NMO11]. De entre os diferentes formatos moleculares destacam-se os seguintes:

- SMILES [Wik17c] - é uma representação das estruturas químicas através de uma linha de carateres, que pode ser importada por grande parte dos editores moleculares.

- *Chemical table file* [Wik17b] - é uma representação mais complexa que descreve as moléculas e as reações químicas, podendo conter informações adicionais, como por exemplo, sinónimos, pubchemID, peso, carga, etc. Para cada molécula é listado cada átomo, as coordenadas xyz do átomo, bem como as ligações entre eles. Da formatação *Chemical table file* destacam-se os formatos MDL e SDF.

Os ficheiros podem ser reduzidos pela filtragem de moléculas segundo um grau de similaridade (coeficiente de Tanimoto), em que valores do coeficiente próximos de 1 correspondem maior parecença entre duas moléculas.

Ferramentas para conversão de formatos

Por vezes é necessário ter um ficheiro de descritores moleculares num determinado formato, tanto por necessidade de informação adicional que o corrente formato não apresenta, como por limitação da ferramenta a usar.

O OpenBabel[NMO11] e o JOELib[Weg01] são duas ferramentas de conversão de formatos, que podem ser executadas por outras aplicações.

Por exemplo,para executar uma conversão do formato **sdf** para o **cml** usando o OpenBabel, pode-se executar a instrução seguinte:

```
1              obabel -i sdf epinephrine.sdf -o cml epinephrine.cml
```

Ficheiros ARFF

O conjunto de moléculas a usar nos algoritmos de extração de conhecimento, deve ser colocado num ficheiro ARFF[oW], formato comum de *input* da ferramenta Weka que vai ser utilizada no trabalho da tese.

Este formato consiste em identificar o nome da relação, a lista de atributos e a lista de instâncias com valores de de cada atributo.

2.2 Descritores moleculares

Um descritor molecular consiste na descrição numérica de uma propriedade da molécula, onde a informação química contida na molécula é codificada num número.

Os descritores moleculares derivam de várias aproximações teóricas, tais como, química orgânica, matemática discreta entre outras. Uma categorização frequente dos descritores moleculares pode ser vista na Table 2.1.

Representação Molecular	Descrição
0D	Peso molecular, número de átomos, número de ligações, soma de propriedades atómicas
1D	Número de fragmentos
2D	Índice de Zagreb, Índice de Wiener, BCUT, vetor de autocorrelação 2D
3D - Descritores Geométricos	Descritores WHIM, vetores de autocorrelação 3D
3D - Propriedades da Superfície	Potencial eletrostático, potencial de hidrofobicidade
3D - Propriedades de grade	Análise comparativa dos campos moleculares

Tabela 2.1: Tipos de descritores moleculares.

Os algoritmos de aprendizagem constroem modelos preditivos através dos atributos das instâncias. Desta forma, a conversão das características das moléculas para valores numéricos, por meio de descritores, é importante para a aplicação dos algoritmos.

Ferramentas para cálculo de Descritores Moleculares

Para calcular os descritores de uma molécula é necessária uma ferramenta que consiga transformar informações contidas dentro da representação simbólica de uma molécula num número útil.

Estas ferramentas recebem como *input* ficheiros que contêm informação química acerca de moléculas e calculam uma série de descritores moleculares.

Uma ferramenta de cálculo de descritores e *open source* é o `PaDEL`[Yap11], que tem a capacidade de calcular 905 descritores moleculares e 10 tipos de impressões digitais (*fingerprints*). Os resultados do cálculo são guardados em ficheiros no formato SMILES ou MDL.

2.3 Extração do conhecimento

Nas últimas décadas, o aumento da complexidade dos problemas a serem tratados computacionalmente e o volume de dados gerados, reforçou a necessidade de desenvolver ferramentas computacionais mais sofisticadas e autónomas, que reproduzissem a necessidade de intervenção humana e a dependência de especialistas.

A introdução de métodos de análise de dados com inferência indutiva, a partir do qual se obtêm conclusões gerais sobre um conjunto de exemplos, permite aprender com a experiência passada. Desta forma, algoritmos de extração de conhecimento aprendem a induzir hipóteses capazes de resolver um problema, a partir de um conjunto de dados que representam instâncias do problema. Os conjuntos de dados a utilizar necessitam porém de uma análise das características de forma a descobrir tendências e padrões que forneçam informações valiosas que ajudem a compreender o processo de geração dos dados. A informação obtida pode ajudar na seleção das técnicas mais apropriadas na etapa de pré-processamento dos dados, bem como para a produção dos modelos preditivos.

Em suma, é imperativo fazer uma análise das características do conjunto de dados antes da aplicação dos algoritmos de extração de conhecimento.

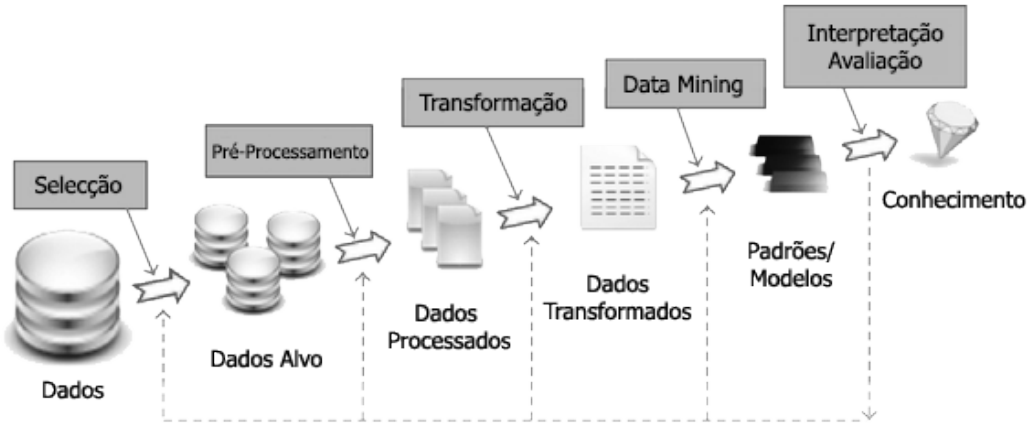

Figura 2.1: Processo de extração de conhecimento.

Análise de Dados

Os métodos de *Data Mining* (exemplo da Figura) podem ser muito valiosos para a resolução do problema em causa.

A definição do tipo de dados a utilizar, assim como a sua escala são fundamentais para fazer uma escolha adequada dos métodos de *data mining* a utilizar.

Os conjuntos de dados são compostos por objetos, também designados por instâncias ou registos, aos quais estão associados a um conjunto de características denominadas de atributos. Desta

forma, cada objeto corresponde a uma linha da tabela de dados e cada atributo a uma propriedade do objeto. Os dados podem ser representados formalmente por uma matriz X_{nxd}, em que n é o número de objetos e d é o número de atributos.

Tipos de dados

Os atributos dos objetos do conjunto de dados podem ser do tipo quantitativo ou qualitativo.

Atributos do tipo quantitativo são representados por valores numéricos que podem ser usados em operações aritméticas. Estes podem ainda estar sub divididos em atributos contínuos ou atributos discretos.

- Atributos contínuos - podem assumir um número infinito de valores, geralmente representados por números reais.
- Atributos discretos - assumem um número finito ou infinito contável de valores.

Por vezes, na extração do conhecimento é necessário saber a medida associada a um atributo para a avaliação do conhecimento. Se existe um atributo relativo ao peso, o valor em si não indica se a medida é o quilograma, grama ou miligrama.

Pré-Processamento

O desempenho dos algoritmos de extração de conhecimento a partir de conjuntos de dados é geralmente afetado pelo estado dos dados.

Os conjuntos de dados contêm objetos de diferentes características, onde os atributos podem apresentar imperfeições ou ruído, com valores incorretos, inconsistentes, duplicados ou ausentes. Além disto os conjuntos de dados podem apresentar um número pequeno ou elevado de objetos, que por sua vez, podem ser caracterizados por um número pequeno ou elevado de atributos.

De forma a melhorar a qualidade dos dados e reduzir/eliminar os problemas supracitados, são utilizadas técnicas de pré-processamento de dados que permitem a construção de modelos mais fiéis e reduzem a complexidade computacional.

Assim, antes da aplicação de algoritmos de extração de conhecimento, devem ser utilizadas técnicas de pré-processamento, tais como, tratamento de dados desbalanceados, técnicas de amostragem, limpeza de dados, integração de dados, transformação de dados e redução da dimensionalidade, a fim de criar modelos mais fiéis, diminuir a complexidade computacional e aumentar a adequabilidade dos dados a um determinado algoritmo.

Remoção Manual de Atributos

Os conjuntos de dados por vezes apresentam atributos que não são necessários para a extração de conhecimento, não fazendo qualquer sentido a sua utilização. Sendo assim, se um atributo não contribui para a avaliação de um atributo alvo, ele é considerado irrelevante e deve ser removido dos objetos.

Integração de Dados

Para dados provenientes de diferentes fontes é necessário recorrer à sua integração antes de aplicar as técnicas de extração de conhecimento.

Neste processo é importante identificar os objetos presentes e realizar uma procura dos atributos comuns nos conjuntos a serem combinados. Por vezes, atributos correspondentes podem apresentar diferentes nomes ou terem sofrido actualizações, o que dificulta a integração dos dados. Para solucionar este entrave utilizam-se metadados.

Amostragem de Dados

A existência de conjuntos de dados grandes não implica, por parte do algoritmo, a utilização de todas as instâncias nele contido, até porque é mais eficiente usar apenas parte dos dados iniciais na maioria das vezes.

Associada à quantidade de dados está a **eficiência computacional** e a **taxa de acerto**, em que o aumento do número de dados leva a uma maior taxa de acerto e a uma menor eficiência computacional, na maioria dos casos. De forma a contrariar esta proporcionalidade entre o custo computacional e a taxa de acerto, recorre-se à construção de uma amostra que permita alcançar o mesmo desempenho do conjunto de dados completo, mas com um custo computacional menor.

A amostra a ser escolhida deve ser representativa do conjunto de dados inicial, obedecendo à mesma distribuição estatística. Para efetuar a recolha da amostra existem três abordagens possíveis:

- **Amostragem Aleatória Simples** - esta abordagem pode ser feita com reposição ou sem reposição, e consiste na escolha aleatória de objetos do conjunto de dados.

- **Amostragem Estratificada** - esta abordagem ocorre quando as classes apresentam propriedades diferentes, por exemplo, uma grande diferença do número de instâncias entre as classes, que pode levar os algoritmos a serem tendenciosos. Esta abordagem pode seguir duas variações: criar uma amostra que mantenha o mesmo número de objetos para cada classe ou criar uma amostra em que o número de objetos de cada classe é proporcional ao número de objetos da classe no conjunto inicial.

- **Amostragem Progressiva** - consiste na construção de uma amostra pequena que vai sendo aumentada progressivamente de tamanho, enquanto a taxa de acerto continuar a melhorar. O grande objetivo desta abordagem é a identificação da menor quantidade de dados necessária de forma a reduzir a perda na taxa de acerto.

Dados Desbalanceados

Em conjuntos de dados desbalanceados o número de objetos varia para as diferentes classes.

A maioria dos algoritmos de extração de conhecimento pioram o seu desempenho na presença de dados desbalanceados, favorecendo a classe maioritária.

É necessária a aplicação de técnicas que tornem o conjunto de dados balanceado, tais como:

- **Redefinir o tamanho do conjunto de dados** - consiste na geração de objetos para a classe minoritária, seguindo o processo de geração do conjunto de dados inicial, ou na redução do número de instâncias da classe maioritária. Esta técnica pode levar a problemas de *underfitting* ou de *overfitting*.
- **Utilizar diferentes custos de classificação para as diferentes classes**
- **Aplicar técnicas de classificação com apenas uma classe.**

Limpeza de Dados

A qualidade dos dados contidos num conjunto é importante para uso das técnicas de extração de conhecimento, pois nem todas conseguem lidar corretamente com imperfeições nos dados.

Associadas a dados imperfeitos podem estar as dificuldades seguintes:

- **ruído** - possuem erros ou valores diferentes do esperado
- **inconsistência** - não combinam ou contradizem outros atributos do mesmo objeto
- **redundância** - quando dois ou mais objetos têm os mesmos valores para todos os atributos, ou dois ou mais atributos têm os mesmos valores para 2 ou mais objetos
- **incompletos** - apresentam a ausência de valores para alguns atributos.

Assim, a melhoria da qualidade dos dados é importante para a remoção de deficiências dos dados que levam a análises e estatísticas incorretas.

2.3.1 Data Mining

O *Data Mining* é um processo de descoberta de padrões em grandes conjuntos de dados envolvendo métodos de ***machine learnng***, **estatísticas** e **sistemas de bases de dados**, e corresponde a uma das atividades do processo de extração de conhecimento. O grande objetivo deste processo é extrair informação de um conjunto de dados e transformá-los numa estrutura para um uso futuro.

Tarefas Data Mining

Classificação - Consiste em identificar qual a classe a que uma molécula pertence. é feita uma análise ao conjunto de moléculas fornecidas, havendo previamente uma atribuição da classe a que pertence cada molécula do conjunto, com a finalidade de "aprender"como classificar a nova molécula[Die98]. Este tipo de aprendizagem é classificada de **supervisionada**.

Regressão - Consiste na construção de modelos para prever os valores numéricos. Também corresponde a uma aprendizagem **supervisionada**.

Clustering - Consiste em formar grupos de muléculas em que as moléculas de cada grupo são "próximas"mas distantes das moléculas dos outros grupos. Um *cluster* é um conjunto de moléculas semelhantes entre si, porém diferentes das moléculas dos outros *clusters*. Esta tarefa difere das anteriores, pois não necessita que as moléculas sejam previamente categorizadas e não pretende classificar, estimar ou prever o valor de nenhuma variável. Esta tarefa apenas identifica os grupos de dados semelhantes. Este tipo de aprendizagem é classificada de **não supervisionada**.

Métodos/Técnicas

As técnicas de *data mining* por *machine learning* são divididas em aprendizagem **supervisionada** (preditivo) e **não supervisionada** (descritivo). Em situações especiais podem ser usada uma abordagem **semi-supervisionada**.

As aprendizagens **supervisionada** e a **não supervisionada** diferem no facto da **não supervisionada** não necessitar de uma pré-categorização para as moléculas, ou seja, não necessita de um atributo alvo. Os métodos que usam aprendizagem **não supervisionada** geralmente usam uma medida de semelhança entre os atributos, como é o caso da tarefa de ***clustering***. Na aprendizagem **supervisionada**, os métodos usam um conjunto de dados que possuem uma variável alvo pré-definida e as moléculas são categorizadas relativamente a essa molécula, como é o caso da tarefa de **classificação**

No processo de *data mining*, diferentes técnicas devem ser testadas e combinadas para determinar a performance de cada técnica e da combinação de técnicas a fim de determinar a combinação de técnicas ou o técnicas a usar. A figura seguinte apresenta um exemplo de combinação de métodos Figura 2.2. As técnicas de **classificação** são usadas para prever valores de variáveis do tipo categóricas.

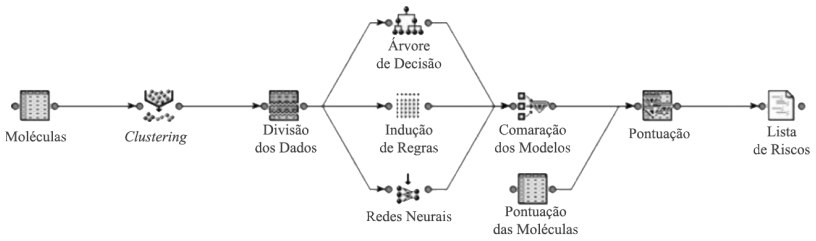

Figura 2.2: Processo de comparação de modelos de diferentes técnicas.

Métodos Baseados em Distâncias

Os métodos baseados em distâncias consideram a proximidade entre os dados nas suas predições, sendo que dados semelhantes tendem a estar concentrados na mesma região do espaço de entrada Figura 2.3.

Um método baseado em distâncias usado com frequência é o algoritmo dos vizinhos mais próximos, considerado o mais simples dos algoritmos de aprendizagem automática. Este algoritmo classifica os novos objetos com base nos exemplos do conjunto de treino que lhe são próximos, através da memorização dos objetos de treino e não pela construção de um modelo compacto para os dados (algoritmo preguiçoso - *lazy*).

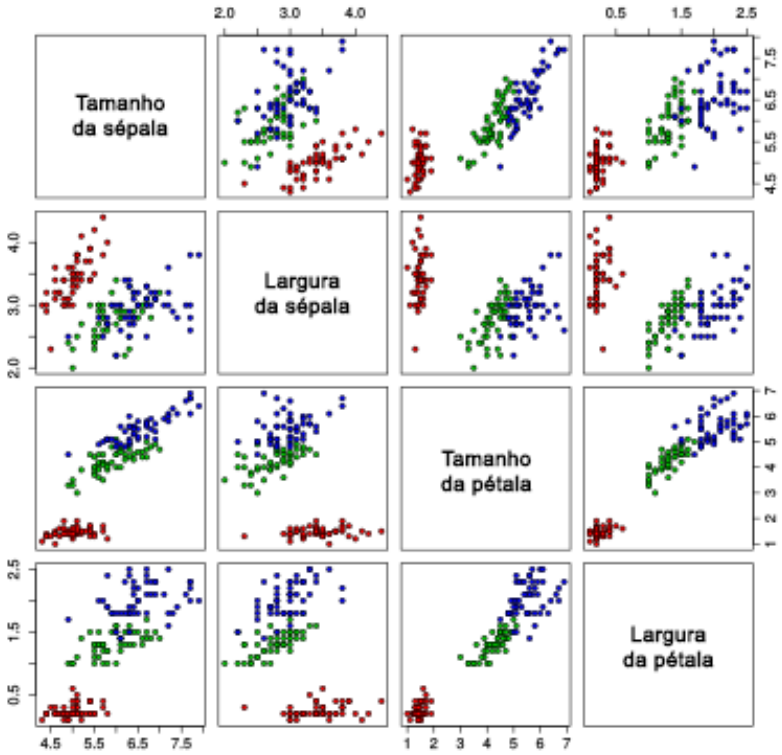

Figura 2.3: Projeção sobre o plano de dois atributos dos objetos do conjunto de dados.

Algoritmo k-NN

Este algoritmo segue o paradigma de que objetos com caraterísticas semelhantes pertencem ao mesmo grupo.

Baseia-se unicamente em memória, sendo toda a computação adiada até à fase de classificação, através da memorização de objetos. Este aspeto torna o algoritmo lento no processo de classificação de exemplos de teste, havendo a necessidade de criar um sub-conjunto de dados refinado. A construção desse novo conjunto de dados pode resultar da eliminação de objetos redundantes ou da eliminação de objetos em que todos os vizinhos são da mesma classe.

Neste algoritmo, os objetos são representados por um ponto no espaço de entrada. A distância entre os pontos do espaço de entrada é calculada através da distância euclidiana. Métrica usualmente usada para este cálculo, entre outras.

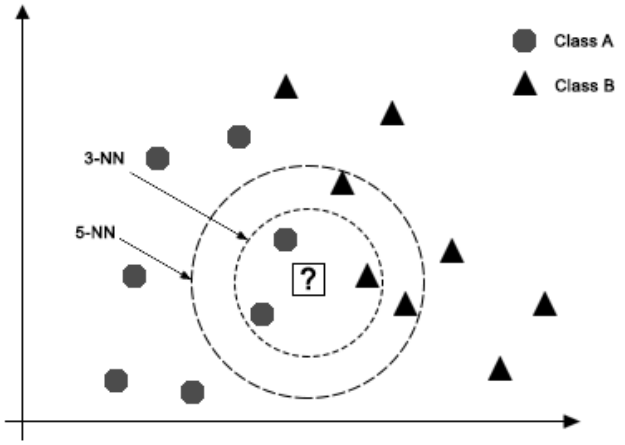

Figura 2.4: Impacto do valor de k no algoritmo k-NN.

Em suma, o algoritmo k-NN é simples e apresenta uma boa taxa de acerto preditiva, sendo bastante influenciado pelo número *k* de vizinhos, como representado na Figura 2.4, e pela medida de distância usada.

2.3.1.1 Métodos Probabilísticos

Os métodos probabilísticos lidam com as tarefas preditivas por meio de algoritmos baseados no **teorema de *Bayes***.

Estes métodos assumem que a probabilidade de um evento A dado um evento B não depende apenas da relação entre eles, mas também da probabilidade de observar A independentemente de observar B[DRO01]. Para isso é feita a observação da frequência com que um evento ocorre de forma a estimar a probabilidade de ocorrer esse evento. Em problemas de decisão o objetivo é estimar a probabilidade de C sabendo que A ocorreu P(C|A), onde C representa a Classe e A o valor observado dos atributos.

Aprendizagem Bayesiana

Os métodos probabilísticos realizam uma aprendizagem bayesiana para a construção dos modelos. Esta aprendizagem baseia-se no teorema de Bayes que separa exemplos de classes diferentes através de funções discriminantes. Sendo c_i uma classe e dado o exemplo **x**, o Teorema de Bayes fornece um método para calcular P(c_i|**x**) :

$$P(c_i|\mathbf{x}) = \frac{P(c_i)P(\mathbf{x}|c_i)}{P(\mathbf{x})} \tag{2.1}$$

O denominador, P(**x**), pode ser ignorado, uma vez que é o mesmo para todas as classes, não afetando os valores relativos das suas probabilidades.

Classificador Naive Bayes

O classificador *Naive Bayes* resume a variabilidade do conjunto de dados em tabelas de contingência, assumindo estas como suficientes para distinguir entre classes.

Assume que os valores dos atributos de um exemplo são independentes entre si para uma classe, *naive*.

2.3.1.2 Métodos Baseados em Procura

Uma árvore de decisão usa a estratégia dividir para conquistar para resolver um problema de decisão.

Esta técnica de classificação consiste num fluxograma em forma de árvore, onde cada nó indica um teste feito sobre um atributo(por exemplo, acidez > 20). As ligações entre os nós representam os valores possíveis do teste do nó anterior, e as folhas indicam a classe(categoria) à qual a molécula em teste pertence.

A criação da **árvore de decisão** é feita através de um conjunto de treino. De seguida usa-se uma heurística para determinar o atributo que melhor diferencia as amostras em classes.

Após montada a **árvore de decisão**, para classificar uma nova molécula, basta seguir o fluxo da árvore mediante os testes nos nós (começando na raiz até alcançar uma folha). Este fluxo caracteriza-se como uma estratégia de divisão e conquista, pois parte de um problema e divide-o em problemas mais simples até alcançar um resultado.

Classificador C4.5

O classificador C4.5 é um método baseado em procura, que constrói árvores de decisão a partir de um conjunto de dados de treinamento utilizando o conceito de Entropia.

Classificador *Random Forest*

O classificador *Random Forest* é um método baseado em procura, que combina a saída de um conjunto de algoritmos de aprendizagem.

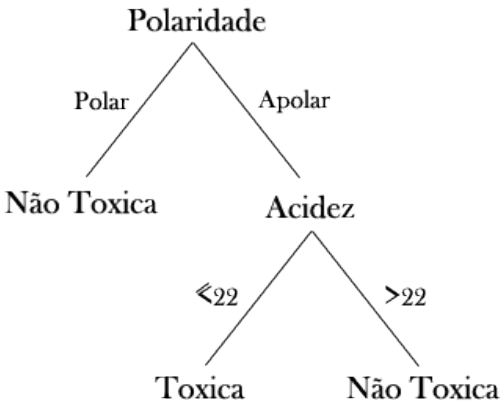

Figura 2.5: Árvore de decisão para um conjunto de moléculas.

São criadas múltiplas árvores de decisão, onde cada árvore é construída independentemente das outras. O resultado da previsão corresponde à votação sobre cada modelo produzido, onde o mais votado é o selecionado.

Estratégias de Poda

A poda é considerada a fase mais importante do processo de construção da árvore, principalmente na presença de dados com ruído, que levam à indução de árvores não confiáveis para classificar novos objetos. Por vezes a árvore induzida tende a ser grande e difícil de compreender, neste caso a troca de nós profundos por folhas ajuda a minimizar esses problemas.

A poda de uma árvore de decisão irá causar certamente a classificação incorreta de alguns exemplos do conjunto de treino. Porém, esta técnica conduz a menores erros de generalização.

2.3.2 Avaliação dos Modelos Preditivos

Na construção de modelos preditivos por algoritmos de extração de conhecimento, o conhecimento é proveniente unicamente do conjunto de exemplos, a partir do qual a indução é realizada. Desta forma devem ser avaliadas as técnicas existentes de modo a escolher a mais adequada ao problema, uma vez que não existe uma técnica universal que obtenha o melhor desempenho para qualquer problema.

Em alguns casos, as caraterísticas das técnicas existentes e do tipo de problema a resolver podem ser consideradas para ajudar na escolha da técnica adequada ao conjunto de dados. Por exemplo, para conjuntos de dados de alta dimensionalidade, o uso de SVMs é o mais adequado, enquanto o algoritmo k-NN, que recorre à distância euclidiana, pode não ser o mais adequado.

Para a avaliação dos modelos preditivos gerados pelas técnicas, são usadas métricas relacionadas com o desempenho obtido nas predições realizadas. Essa avaliação é normalmente realizada por meio da análise do desempenho na classificação de novos exemplos, que não foram utilizados no conjunto de treino[eBJA03].

Métricas para Classificação

Na avaliação do desempenho de um classificador $\hat{f}$ uma das métricas usadas é a taxa de erro ou taxa de classificações incorretas, ilustrada na Equação 2.2. Esta taxa é equivalente à proporção de exemplos do conjunto classificados incorretamente pelo classificador $\hat{f}$. Sendo n o número de exemplos.

$$err(\hat{f}) = \frac{1}{n}\sum_{i=1}^{n} I(c_i \neq \hat{f}(\mathbf{x}_i)) \tag{2.2}$$

Em que $I(a) = 1$ se a é verdadeiro e 0 se caso contrário. A taxa de erro calculada varia entre $[0,1]$, onde os valores próximos de 0 são os melhores. O complemento corresponde à taxa de acerto do classificador, sendo os valores próximos de 1 considerados melhores.

A avaliação do desempenho de um classificador pode também ser analisada através do uso de uma matriz de confusão, que ilustra o número de predições corretas e incorretas para cada classe Figura 2.6. As linhas dessa matriz representam as classes verdadeiras, e as colunas representam as classes preditas. Assim sendo, cada elemento m_{ij} da matriz de confusão apresenta o número de exemplos da classe i classificados como sendo da classe j. A dimensão dessa matriz é igual ao quadrado do número de classes usadas para a classificação.

	$ClasseA'$	$ClasseB'$
$ClasseA$	30	2
$ClasseB$	7	25

Figura 2.6: Matriz de confusão para um problema de duas classes.

A análise desta matriz permite inferir quais as classes em que o algoritmo apresenta maior dificuldade em discriminar.

Amostragem

Na criação de um modelo preditivo devem ser usadas amostras diferentes para a indução e a avaliação do modelo, uma vez que a utilização de exemplos usados na fase de treino produz estimativas otimistas.

Para obter estimativas de desempenho preditivo mais estáveis, devem ser usados métodos de amostragem para definir os sub-conjuntos de treino e teste a usar. Os dados de treino serão utilizados na indução e no ajuste do modelo, enquanto os dados de teste serão usados para avaliar o classificador. Nenhum sub-conjunto de exemplos do conjunto de teste deve estar presente no conjunto de treino, de forma a assegurar que estes não foram usados na indução do modelo.

A criação das amostras para treino e teste podem ser feitas através de diferentes métodos:

- *Holdout* é um tipo de amostragem que divide um conjunto de dados em dois sub-conjuntos, treino e teste, normalmente com a proporção $\frac{2}{3}$ para treino e $\frac{1}{3}$ para teste.

- *Crosss-Validation* é um método que divide o conjunto de exemplos em *r* subconjuntos de tamanho aproximadamente igual, onde os objetos de *r-1* partições são utilizados no treino do modelo. Este método é repetido *r* vezes, sendo utilizada em cada iteração uma partição diferente para teste.
- *Bootstrap* é um método que gera *r* conjuntos de treino a partir de um conjunto de exemplos original. Os exemplos são amostrados aleatoriamente com reposição, podendo estar presentes num subconjunto de treino mais de uma vez. Os exemplos não selecionados compõem os subconjuntos de teste.

Análise no Espaço ROC

A análise de classificadores em problemas de duas classes pode ser feita através da análise entre a taxa de verdadeiros positivos(VP) e a taxa de falsos positivos (FP), sendo uma classe denotada de positiva (+) e outra denotada de negativa (-). A matriz de confusão é ilustrada na Figura 2.7.

- **VP** - corresponde ao número de exemplos da classe positiva classificados corretamente.
- **VN** - corresponde ao número de exemplos da classe negativa classificados corretamente.
- **FP** - corresponde ao número de exemplos da classe negativa classificados incorretamente.
- **FN** - corresponde ao número de exemplos da classe positiva classificados incorretamente.

	$Classe A(+)$	$Classe B(-)$
$Classe A(+)$	VP	FN
$Classe B(-)$	FP	VN

Figura 2.7: Matriz de confusão para um problema de duas classes.

Medidas de Desempenho

A partir da matriz de confusão, para *n* exemplos do conjunto de dados, da Figura 2.7 obtêm-se a medidas de desempenho seguintes[eBJA03]:

- *Taxa de erro na classe positiva*: taxa de exemplos da classe positiva incorretamente classificados pelo classificador $\hat{f}$. Taxa de falsos negativos (TFN).

$$err_+(\hat{f}) = \frac{FN}{VP+FN} \tag{2.3}$$

- *Taxa de erro na classe negativa*: taxa de exemplos da classe negativa incorretamente classificados pelo classificador $\hat{f}$. Taxa de falsos positivos (TFP).

$$err_-(\hat{f}) = \frac{FP}{FP+VN} \tag{2.4}$$

- *Taxa de erro total*:
$$err(\hat{f}) = \frac{FP+FN}{n} \tag{2.5}$$
- *Taxa de acerto*:
$$ac(\hat{f}) = \frac{VP+VN}{n} \tag{2.6}$$
- *Precisão*: taxa de exemplos positivos classificados corretamente entre todos os preditos como positivos.
$$prec(\hat{f}) = \frac{VP}{VP+FP} \tag{2.7}$$
- *Sensibilidade*: taxa de acerto na classe positiva, ou taxa de verdadeiros positivos (TVP)
$$sens(\hat{f}) = \frac{VP}{VP+FN} \tag{2.8}$$
- *Especificidade*: taxa de acerto na classe negativa.
$$esp(\hat{f}) = \frac{VN}{VN+FP} \tag{2.9}$$
O complemento da *especificidade* corresponde à taxa de falsos positivos (TFP).
$$TFP(\hat{f}) = 1 - esp(\hat{f}) \tag{2.10}$$

Análise ROC

A performance de classificadores em problemas de duas classes pode ser analisada recorrendo a espaços ROC (*Receiving Operating Characteristics*)[Faw05].

Um espaço ROC é constituído por dois eixos X e Y que representam as taxas de falsos positivos e verdadeiros positivos, respetivamente. O desempenho de um classificador é representado por um ponto neste espaço. A Figura 2.8 ilustra um espaço ROC.

A linha diagonal representa classificadores que realizam predições aleatórias. Desempenhos de classificadores assinalados abaixo desta linha são considerados piores que aleatórios. O ponto (0,1) representa todas classificações corretas, em que todos os exemplos positivos e negativos são classificados corretamente (*ROC Heaven*), em oposição ao ponto (1,0) que representa o pior cenário (*ROC Hell*).

Um ponto deste espaço é considerado melhor que outro se tiver posicionado mais acima e à esquerda que o segundo.

2.3.3 Ferramentas de Data Mining

2.3.3.1 Weka

É uma ferramenta livre e possui uma série de algoritmos para diferentes tarefas de *data mining* [EF16, MH]. Os algoritmos podem ser aplicados diretamente na ferramenta, ou utilizados por

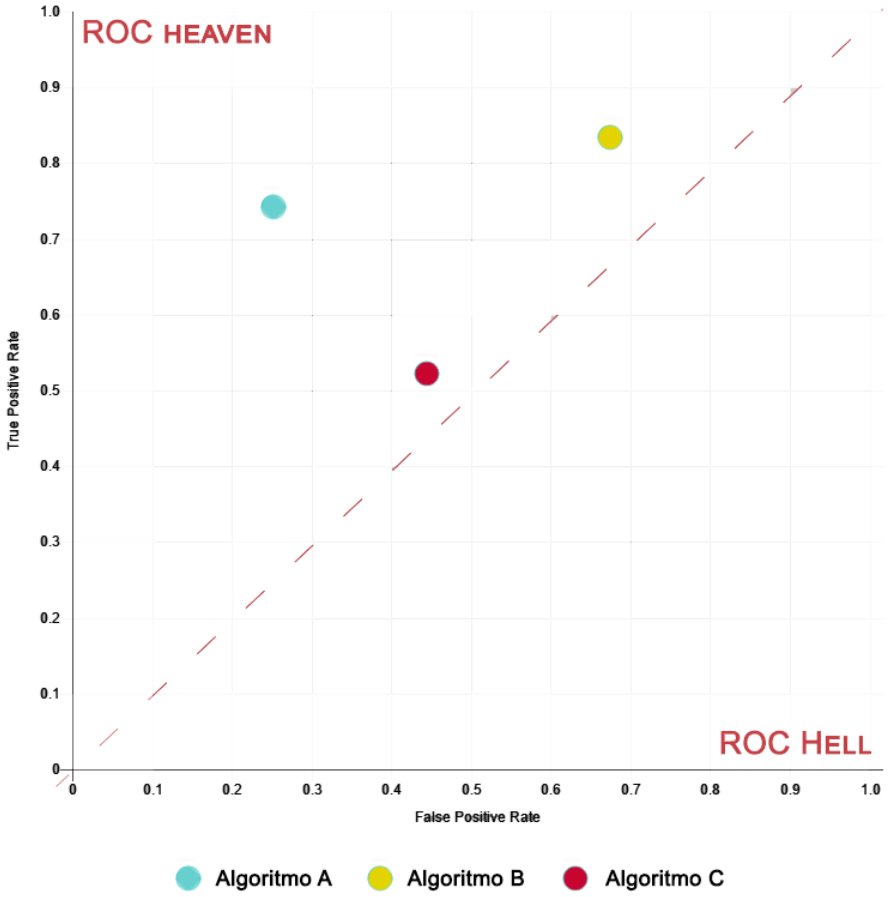

Figura 2.8: Espaço ROC com três classificadores.

outros programas. Fornece as funcionalidades para pré-processamento, classificação, regressão, agrupamento, regras de associação e visualização.

2.3.3.2 Programação lógica indutiva (PLI)

A **programação lógica indutiva** é um sub-campo da *machine learning* que usa a lógica como uma representação uniforme para hipóteses. Para um dado conjunto de exemplos a PLI vai derivar uma hipótese, sendo considerada a interseção entre a *machine learning* e a programação lógica.

Em suma, a PLI deriva regras a partir de exemplos, sendo desta forma uma boa abordagem para problemas de predição e classificação. Isto porque consegue induzir hipóteses a partir de exemplos e desta forma obter conhecimento.

2.3.3.3 Rapidminder

O Rapidminder é um software de extração de conhecimento, análise preditiva e análise de negocio. Ele é usado para investigação, educação, formação, prototipagem e desenvolvimento de

aplicativos, suporta todas as etapas do processo de *data mining* de dados, incluindo a preparação de dados, resultados de visualização , validação e otimização.

2.4 Repositório de Dados Moleculares

2.4.1 PubChem

O PubChem é um repositório de dados de moléculas, disponível online, de acesso gratuito através da interface do site principal ou da API *rest* fornecida.

Neste repositório de dados existe informação de milhões de moléculas e de ensaios sobre conjuntos de moléculas. Os ensaios correspondem à realização de testes sobre um conjunto de moléculas para um alvo estipulado.

Através da API é possível obter conjuntos de moléculas e ensaios no formato JSON ou XML.

Sumário do Capitulo

Neste capitulo, foram descritos os métodos e técnicas usados para a extração de conhecimento de conjuntos de dados, bem como a descrição das ferramentas a usar na aplicação desenvolvida.

Capítulo 3

Estrutura da Aplicação

A aplicação desenvolvida nesta tese teve como foco principal a integração de diversas funcionalidades e ferramentas úteis para o desenho de novos fármacos. Sendo estas funcionalidades e ferramentas abstraídas por interfaces simples e de fácil interação.

Para a integração das diversas funcionalidades e ferramentas numa aplicação multi plataforma, desenvolveu-se a aplicação com tecnologias Web (*WEB aplication*). Esta escolha permite ter uma aplicação que corra em várias plataformas (Windows, macOS, Linux), permite a criação de módulos e serviços para lidar com os componentes externos e internos, a fácil criação de vários processos e tarefas assíncronas sem bloqueio, e uma fácil divisão e manipulação da interface do cliente.

Foi escolhido o *design pattern* MCV (Modelo-Vista-Controlador) para dividir a aplicação em três partes distintas e interligadas. A interligação das três partes é gerida pelo AngularJS.

Neste capitulo é descrita a estrutura da aplicação desenvolvida, começando pela identificação dos principais componentes, seguido da divisão interna dos componentes e interações, tipo de armazenamento de dados e da identificação das etapas de extração de conhecimento.

3.1 Componentes da aplicação

A aplicação foi desenvolvida segundo o *design pattern* MCV, estando desta forma dividida em **Vistas**, **Modelos** e **Controladores**.

Para gerir esta estrutura escolhi usar o AngularJS, que interliga as partes através da injeção de conteúdo e associação de variáveis entre as vistas e os controladores. O AngularJS cria um estado para cada vista, ao qual é associado um controlador que faz a gestão do conteúdo da vista. A transição entre vistas corresponde a uma transição de estados, dando a sensação ao Utilizador de estar a usar uma aplicação de uma única página, uma vez que nunca ocorre um *load* total da página.

As **vistas** são estruturadas em HTML e o estilo foi desenvolvido em CSS3. Na estrutura das vistas existem variáveis que o Angular vai interpretar e associar a variáveis do controlador.

Os **controladores** são o intermediário entre as vistas e os serviços. Um serviço fornece diferentes funcionalidades, nomeadamente a comunicação com os **modelos**. Nesta aplicação os serviços têm instalados módulos para a comunicação com a base de dados, criação e execução de processos, criação de *requests* HTTP, criação de tarefas assíncronas e tarefas de gestão do sistema de ficheiros.

O controlador da vista de criação de modelos preditivos tem funcionalidades extra para ajudar na avaliação e escolha do algoritmo a usar. É usado o módulo *ChartJS* para a análise gráfica das taxas de acerto de diferentes algoritmos e para a análise da relação Verdadeiro Positivo - Falso Positivo de várias instâncias dos algoritmos. Na segunda análise gráfica, relação VP-FP, foram adicionadas funcionalidades extra, para permitir uma melhor interpretação dos resultados por parte do Utilizador.

São identificados de seguida os serviços implementados na aplicação e as funcionalidades em cada um.

3.1.1 Serviços

Source Service

- ***addSource*** - criar uma nova fonte.
- ***removeSource*** - remover uma fonte.
- ***getSources*** - obter todas as fontes.

Original Dataset Service

- ***loadAID (REQUESThttp)*** - esta funcionalidade realiza um *request* HTTP ao Pubchem para obter informação de um ensaio no formato JSON.
- ***createOriginalDataset (requestHTTP)*** - cria um novo conjunto de dados a partir de um ficheiro de moléculas. Para cada molécula do ficheiro, caso não exista ainda na BD, esta é adicionada á base de dados. No final associa ao conjunto de dados criado as moléculas criadas ou selecionadas.
- ***getOriginalDatasets*** - obter os conjuntos de dados existentes.

Test Dataset Service

- ***createTestDataset*** - cria um conjunto de dados de teste a partir de um conjunto de dados filtrado pelo coeficiente de Tanimoto.
- ***applyTanimotoFilter (OpenBabel)*** - cria uma amostra a partir da filtragem de moléculas semelhantes de um conjunto de dados. Este serviço cria processos para a conversão das

moléculas para o formato SMILES, seguidos da execução de um processo que processa, de forma aleatória, a escolha e comparação entres as moléculas do conjunto de dados. No final do processo os ficheiro temporários são eliminados.

- ***removeMolecules*** - remove uma molécula do conjunto de dados de teste.
- ***getDatasetTypes*** - obter os tipos de conjunto de dados.

Descriptor Service

- ***calculateDesc (PaDEL)*** - calcula os descritores moleculares para cada molécula de um conjunto de dados. São lançados processos para o cálculo dos descritores, sendo definido um numero máximo de moléculas por processo para acelerar o mesmo. No final de cada processo os descritores são armazenados e associados às respetivas moléculas.

Project Service

- ***createProjet*** - criar um novo projeto.
- ***getProject*** - obter os dados de um projeto.
- ***getProjects*** - obter os projetos de um utilizador.

Experience Service

- ***createWekaExperience*** - criar uma nova experiência.
- ***createWekaModel*** - criar um novo modelo.
- ***createARFF (WEKA)*** - criar um conjunto de ficheiros ARFF. É criado um ficheiro ARFF para as moléculas e os descritores identificados. A partir do ficheiro criado são lançados dois processos para criar dois conjuntos de dados com instâncias diferentes e dimensões diferentes (conjunto de dados de treino e de teste).
- ***createModelFile (WEKA)*** - criar o ficheiro com o modelo preditivo. É executado um processo para a criação do ficheiro.
- ***validateARFF (WEKA)*** - analisar um ficheiro ARRF. É executado um processo que corre o ficheiro ARFF e analisa para cada atributo o numero de instâncias com valores únicos, distintas, com valores em falta e tipo dos atributos.
- ***getWekaExperiences*** - obter as experiências de um projeto.
- ***getWekaExperience*** - obter a informação de uma experiência.
- ***getModels*** - obter os modelos de uma experiência.

- ***runAlgorithm (WEKA)*** - correr um algoritmo de classificação. É executado um processo para a execução do algoritmo.
- ***runModel (WEKA)*** - correr um modelo e obtém a lista de moléculas testadas e as suas predições.

Na Figura 3.1 está esquematizada a interação entre os componentes da aplicação.

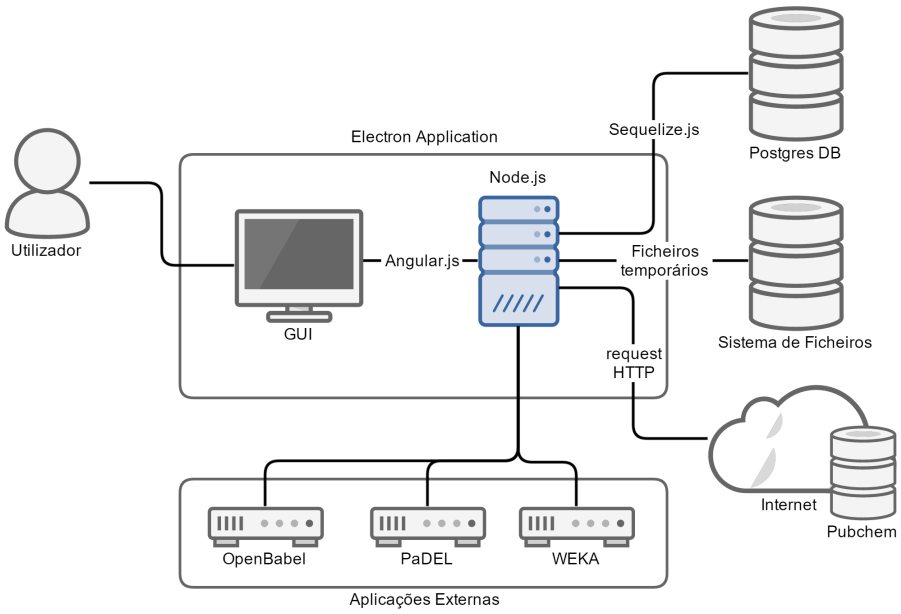

Figura 3.1: Estruturação dos componentes da aplicação.

3.2 Armazenamento de dados

Para gerir o armazenamento dos dados é usado o módulo *SequelizeJS* que realiza o mapeamento objeto-relação para *PostgresSQL*. Para o armazenamento dos ficheiros temporários é usado o módulo *fs* (*file system*) do *NodeJS*. Os ficheiros temporários são criados para o cálculo de descritores, aplicação do filtro Tanimoto, execução do modelo preditivo e execução dos algoritmos.

A estrutura da base de dados está representada na Figura 3.2 .

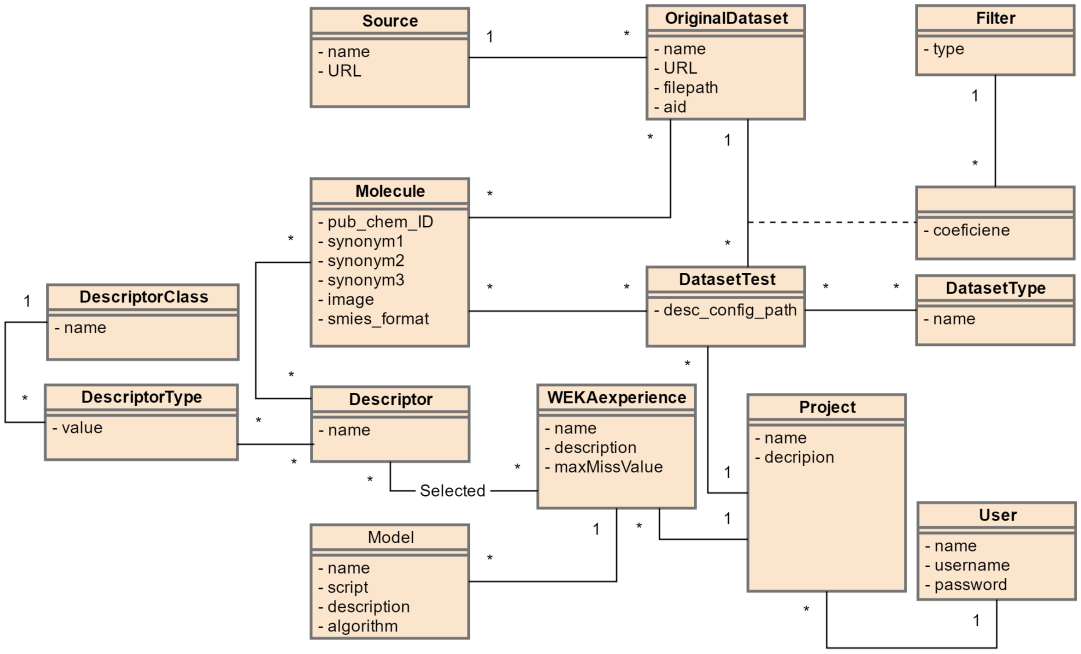

Figura 3.2: Estrutura da base de dados.

3.3 Etapas de extração de conhecimento

Nesta secção são identificadas as etapas de extração de conhecimento existentes na aplicação.

3.3.1 Seleção

A **seleção** do conjunto de dados é feita no ato de criação de um novo *dataset*, onde o Utilizador fornece um conjunto de moléculas à aplicação através de um ficheiro carregado ou de um ID de um ensaio existente no **PubChem**. Esse conjunto de moléculas selecionado será armazenado no sistema e ficará disponível a ser usado num novo projeto.

3.3.2 Pré-Processamento

O **pré-processamento** dos dados está presente na aplicação desde a fase de criação de um projeto novo até à fase de criação de uma nova experiência.

3.3.2.1 Integração

Na etapa de criação de um novo *dataset*, podem ser carregados ficheiros com diferentes formatos e diferentes informações.

De modo a uniformizar a informação a guardar para cada molécula, procede-se à conversão do ficheiro para o formato SDF. De seguida, de modo a garantir que todos os atributos necessários a guardar para cada molécula estão presentes, efetua-se um pedido http a um **armazém de dados** (PubChem) a requerer a informação de um conjunto de moléculas. Esse pedido é efetuado enviando a estrutura SMILES das moléculas, ou o PubChemID.

Após a etapa anterior é necessário proceder ao cálculo dos descritores de cada molécula, a fim de obter as caraterísticas que servirão de *input* aos algoritmos de extração de conhecimento.

Após estas etapas, conclui-se a integração dos dados passando-se para as fases seguintes de **amostragem** e **limpeza** dos dados a utilizar numa experiência.

3.3.2.2 Amostragem e Limpeza

A *amostragem* e a *limpeza* dos dados são dois processos muito importantes para a criação de dados de teste com qualidade.

Na fase de criação de um novo projeto, após escolher o *dataset* original e dar as informações básicas do projeto, é pedido ao Utilizador para indicar o coeficiente Tanimoto a usar no conjunto de dados selecionado. Este coeficiente vai ser usado para filtrar as moléculas do conjunto de dados segundo o grau de parecença entre elas. Neste processo estão presentes tanto a **amostragem** como a **limpeza** dos dados, pois é feita uma **remoção de moléculas segundo um grau de parecença (redundantes)** e ao mesmo tem esta a ser **criada uma amostra representativa do *dataset* original**(*dataset* de teste).

Após criar um novo projeto o Utilizador tem acesso a todos os detalhes do *dataset* de Teste onde pode **remover manualmente moléculas do *dataset*** de forma a reduzir a amostra.

A fase de criação de uma nova experiência é a última fase onde ocorrem os processos de **amostragem** e **limpeza** dos dados. Nesta fase o Utilizador escolhe os descritores a usar na experiência e é criado um ficheiro ARFF com os dados a passar aos algoritmos. Sobre este ficheiro criado ocorrem os processos seguintes:

- filtragem do atributo puchemID que é irrelevante, pois não contribui para a estimativa do valor atributo alvo. Apenas será usado após a previsão dos algoritmos, para identificar as moléculas.

- remoção dos atributos com uma **taxa de de valores em falta** superior à dada pelo utilizador.

- divisão da amostra atual em duas amostras com objetos diferentes e dimensões diferentes: amostra de treino que corresponde a 2/3 da amostra atual e amostra de teste que corresponde a 1/3 da amostra atual.

Esta é a última fase onde os dados são processados, resultando duas amostras que serão usadas na criação dos modelos preditivos.

3.3.3 *Data Mining* e Avaliação de Modelos

O processo de *data mining* e a avaliação de modelos ocorrem em simultâneo na fase de criação de um novo modelo. Nesta fase o Utilizador corre vários algoritmos com diferentes opções ao mesmo tempo que lhe é fornecida a informação do ultimo algoritmo que correu e a informação acumulada do desempenho dos algoritmos corridos.

De seguida são apresentados os algoritmos disponíveis nesta fase e as opções fornecidas para os mesmos.

J48

Classificador baseado em pesquisa, que gera uma árvore podada ou por podar.

- *Coeficient Factor* - Coeficiente usado na poda.
- *MinNum per Leaf* - Numero mínimo de instâncias por folha.
- *Seed* - Variável usada para na aleatoriedade do processo de redução do erro de poda.
- *Num Folds* - determina a quantidade de dados usados para a redução do erro de poda.

As duas última opções apenas são usadas no modo de redução de erro na poda.

Random Forest

Classificador baseado em pesquisa, constrói uma floresta de árvores aleatórias.

- *Max Depth* - profundidade máxima das árvores(0 para sem limite).
- *Num Features* - numero de atributos a serem usados na seleção aleatória.
- *Num Trees* - numero de árvores a gerar.
- *Seed* - numero usado na aleatoriedade.

IBK

Classificador baseado em distância, k-NN (K vizinhos mais próximos).

- *KNN* - numero de vizinhos a usar.
- *Window Size* - numero maximo de instâncias permitidas no conjunto de treino.
- *Distance Weighting* - método de ponderação de distância usado.
  - *no distance*
  - *Weigh by 1/distance*
  - *Weigh by 1-distance*

Este classificador pode ser usado no modo *cross validate*, que vai selecionar o melhor k.

KStar

K * é um classificador baseado em instâncias, ou seja, a classe de uma instância de teste é baseada na classe das instâncias de treino semelhantes a ela.

- *Global Blend* - parâmetro para o *global blending*.
- *Missing Mode* - determina como são tratados os valores em falta.
  - *Ignore the instances with missing values*
  - *Treat missing values as maximally different*
  - *Normalize over the attributes*
  - *Average column entropy curves*

Este classificador pode ser usado no modo *entropic auto blend*, caso o *blending* deva ser baseado em entropia.

NaiveBayes

Classificador baseado em probabilidades.

Este classificador pode ser usado no modo *use kernel estimator*, de forma a usar o *kernel estimator* para os atributos numéricos em vez de uma distribuição normal.

Logistic

Algoritmo de regressão.

- *Max Iterations* - numero máximo de iterações a realizar.

SMO

Classificador baseado em otimização, implementa o algoritmo de otimização mínima sequencial de John Platt para treinar um classificador de vetores de suporte.

- *c* - parâmetro de complexidade.
- *Filter Type* - determina como/se os dados vão ser transformados.
  - *Normalize training data*
  - *Standardize training data*
  - *No normalization/standardization*

Este classificador pode ser usado no modo *build logistic models*, para estimativas de probabilidade adequadas.

Sumário do Capitulo

Neste capitulo foi apresentada de forma detalhada os componentes e a implementação da aplicação, sendo também feita a identificação dos métodos e processos de extração de conhecimento.

Em suma, foi desenvolvida uma aplicação Web multi-plataforma, que junta diferentes ferramentas essenciais no desenho relacional de fármacos. A aplicação estabelece comunicação com o PubChem para a obtenção de dados relativos a moléculas ou ensaios, converte diferentes tipos de ficheiros de moléculas, calcula descritores moleculares de 3 categorias (Fingerprints, 1D_2D e 3D), cria modelos preditivos para diferentes algoritmos, fornece interfaces para avaliar os modelos gerados e a possibilidade de testar moléculas desconhecidas através de um modelo preditivo gerado.

Capítulo 4

Interfaces e Navegação

As vistas da aplicação estão agrupadas em três grupos abstratos: *home*, projeto e experiência, sendo cada grupo uma vista abstrata.

Cada uma das vistas abstratas contem vistas secundárias que são carregadas no seu corpo, sem que ocorra um *reload* total da página. Desta forma, o Utilizador tem uma experiência fluida dos conteúdos. As três vistas têm em comum as barras de navegação superior e lateral esquerda, sendo o espaço central a zona onde as sub vistas serão carregadas.

As vistas são consideradas pelo *angular-ui-router* como estados, por conseguinte as transições entre vistas são consideradas mudanças de estado. Em certas transições de estado existe a passagem de argumentos, nomeadamente na abertura de um projeto (transição do estado home.projeto -> projeto.details) e na abertura de uma experiência (projeto.wekaExperience -> experience.models), pois é necessário informar o estado seguinte do objeto a ser carregado.

Neste capitulo serão apresentadas as vistas da aplicação, bem como as interações que o Utilizador pode realizar nestas.

4.1 Mapa de Navegação

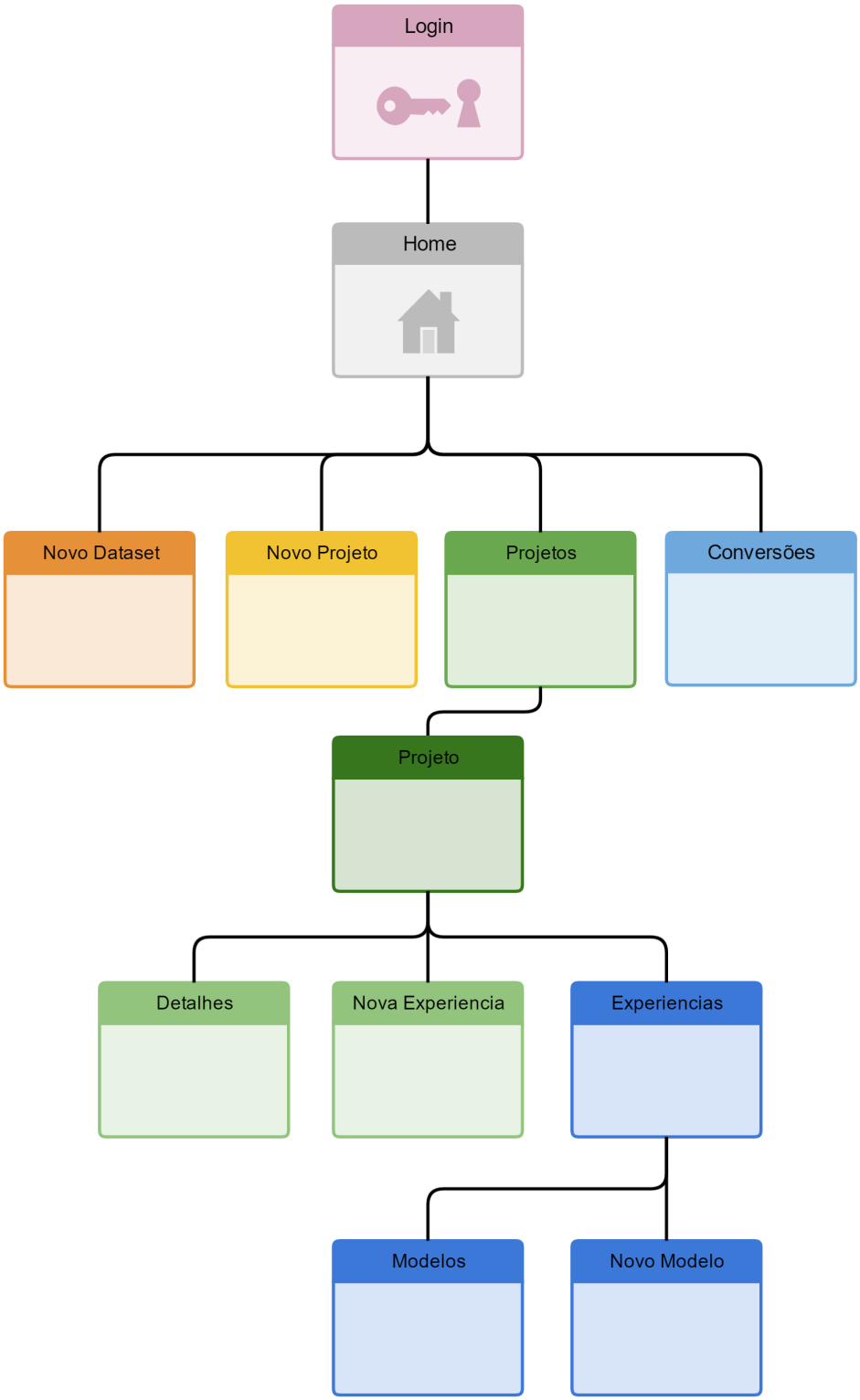

Figura 4.1: Mapa de navegação entre as vistas.

4.2 Login

A vista do *login* não está associada a nenhum dos três grupos abstratos. Aqui o utilizador realiza a sua autenticação, para aceder às funcionalidades da aplicação Figura 4.2. Após a autenticação é realizada a transição para o estado (home.novoDataset).

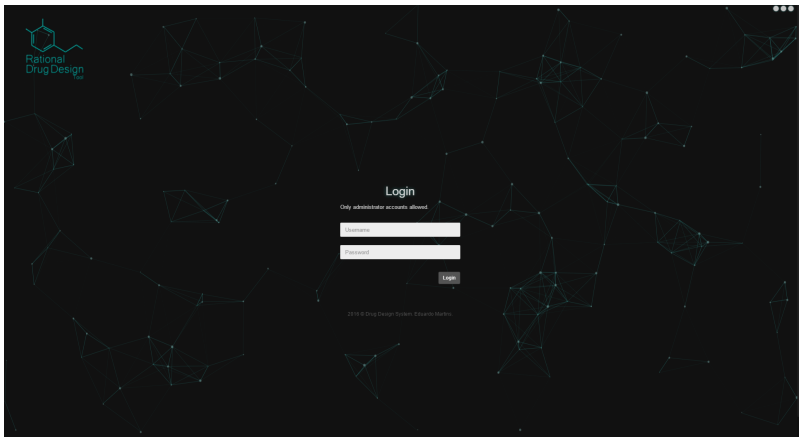

Figura 4.2: Página de Inicio de Sessão.

4.3 Home

A vista *home* é uma vista abstrata, sobre a qual podem ser carregadas quatro vistas: **novoDataset**, **novoProjeto**, **projetos**, **conversão**.

O Utilizador pode fazer a transição entre as sub-vistas da *home* através da barra lateral e terminar a sessão, a qualquer momento, através da barra superior.

4.3.1 Novo Dataset

Nesta vista é apresentado um formulário para a criação de um novo conjunto de dados, sendo este processo dividido em três fases. A transição entre as fases é feita após a validação dos campos da corrente fase.

Na primeira fase o utilizador indica a fonte dos dados a utilizar, sendo possível escolher uma já existente, através da listas de fontes indicada, ou criar uma nova fonte Figura 4.3 .

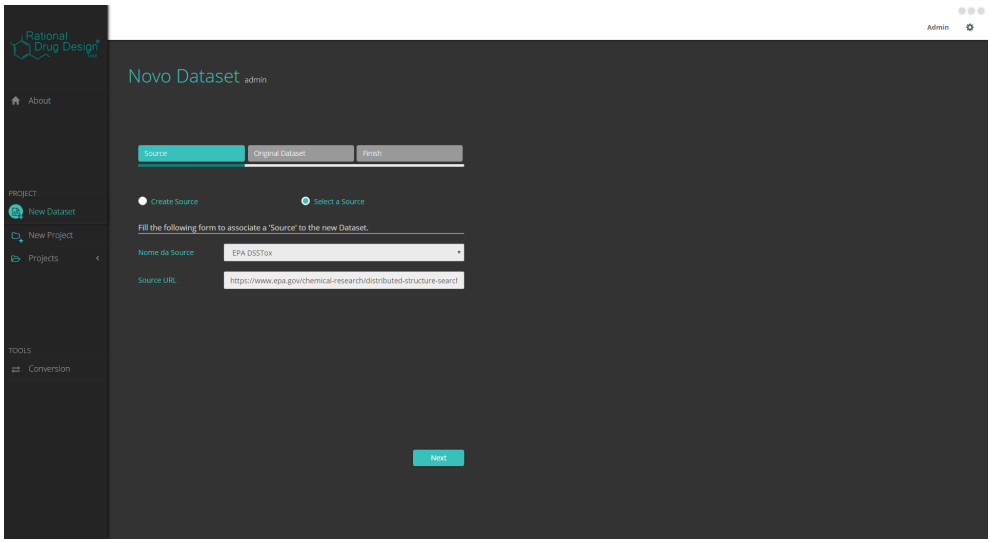

Figura 4.3: Escolha ou criação da fonte dos dados.

Na fase seguinte, é pedido ao Utilizador para indicar o ID do ensaio Pubchem. Após indicar o ID e iniciar o *load* irá ser feito um pedido desse ensaio ao Pubchem, que quando concluído irá automaticamente preencher os restantes campos Figura 4.4. O Utilizador tem também a possibilidade de carregar um ficheiro local, mas sem a garantia de ter informação extra relativa ao conjunto de dados e às moléculas. A informação extra corresponde à descrição dos testes que foram feitos para a construção do conjunto de dados, o protocolo usado e comentários.

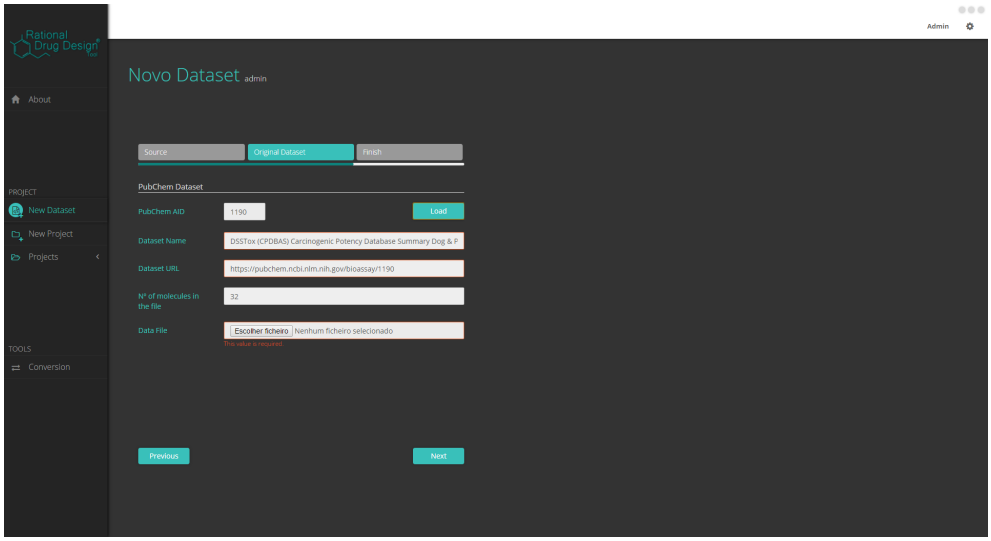

Figura 4.4: Escolha do ensaio Pubchem ou carregamento de um ficheiro local.

Na ultima fase, o utilizador pode confirmar os dados de criação do projeto e iniciar o processo de armazenamento do novo conjunto de dados. Neste processo será estabelecida uma comunicação com o Pubchem para obter as moléculas em falta na base de dados Figura 4.5, e de seguida procede ao calculo dos descritores e armazenamento das moléculas Figura 4.6.

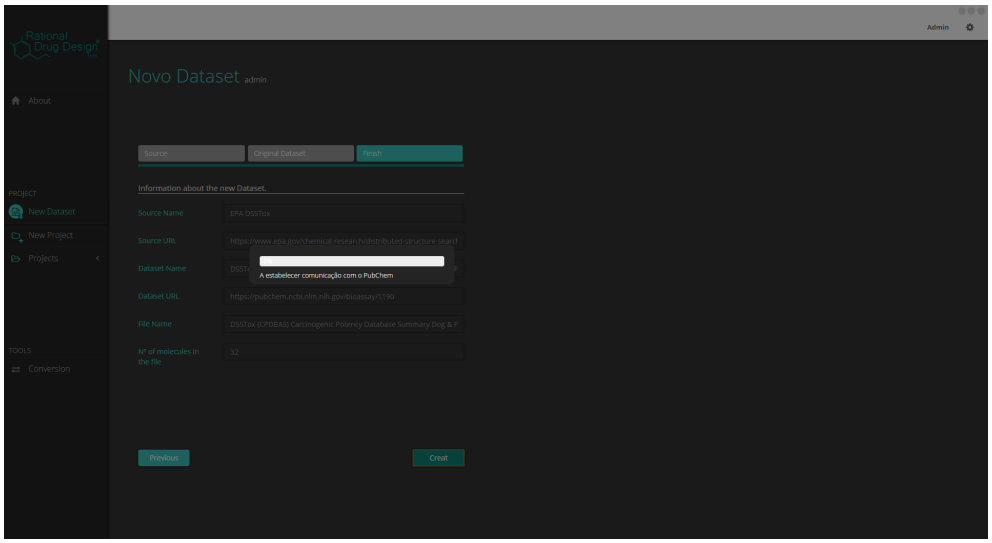

Figura 4.5: Estabelecer comunicação com o Pubchem.

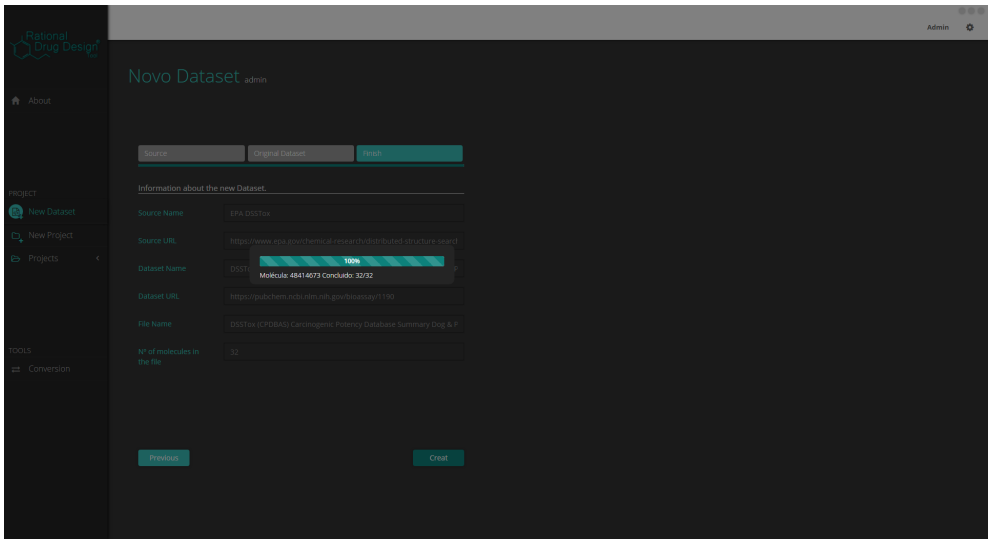

Figura 4.6: Calcular e armazenar os descritores e a informação de cada molécula.

No final do processo de criação do novo conjunto de dados, será feita a transição para a vista de criação de um novo projeto.

4.3.2 Novo Projeto

Nesta vista é apresentado um formulário para a criação de um novo projeto, sendo este processo dividido em cinco fases. A transição entre as fases é feita após a validação dos campos da corrente fase.

Na primeira fase é pedido ao Utilizador para indicar o nome e a descrição do projeto, não sendo possível o uso de nomes para o projeto em uso por esse utilizador Figura 4.7.

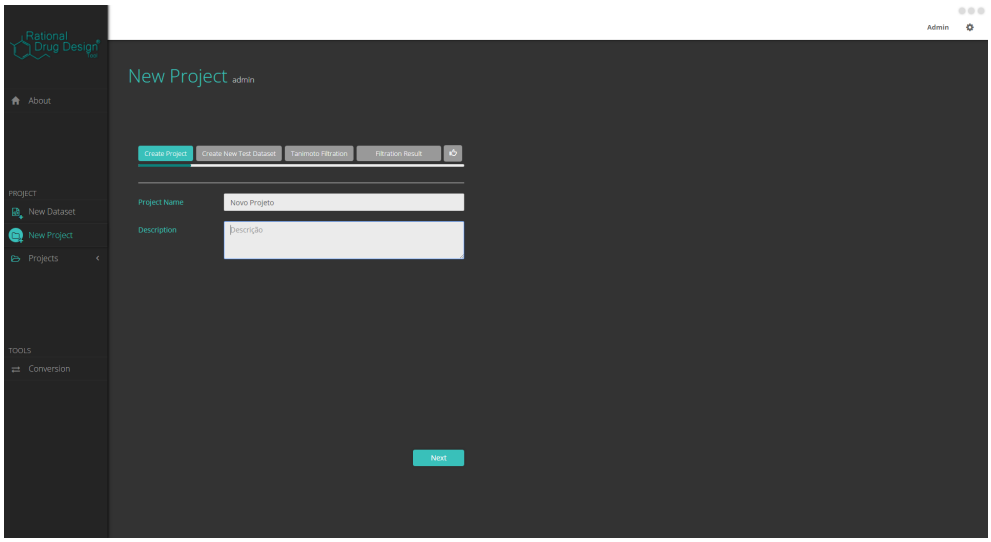

Figura 4.7: Formulário para a criação de um novo projeto.

De seguida o Utilizador tem um formulário para a criação do conjunto de dados de teste a ser usado. Nesta fase o Utilizador deve indicar o conjunto de dados a usar no projeto, Figura 4.8, e o tipo de dados que vai analisar Figura 4.9.

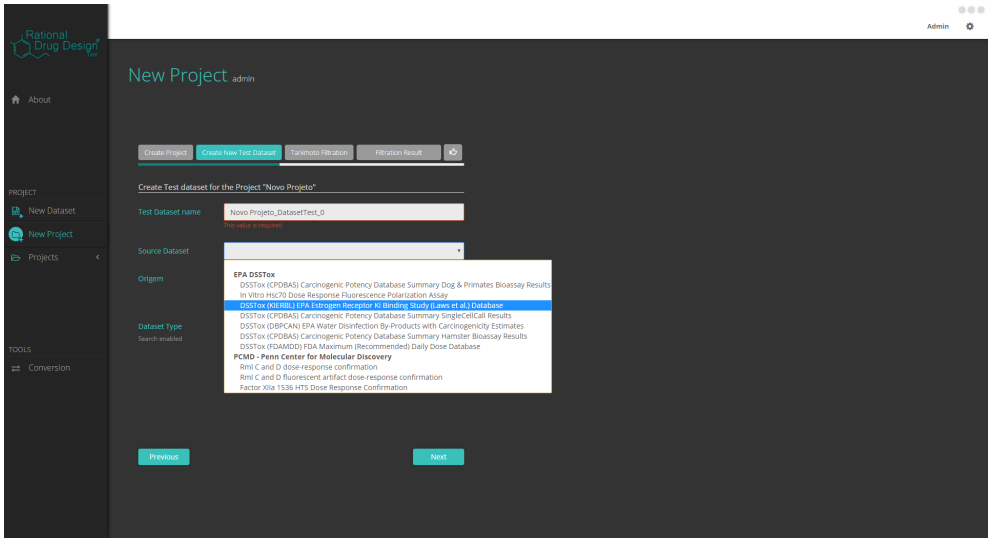

Figura 4.8: Seleção do conjunto de dados a utilizar.

Após a identificação do conjunto de dados a usar, o Utilizador passa para uma fase seguinte onde se iniciará o pré-processamento do conjunto de dados original. Nesta fase o Utilizador escolhe o coeficiente de Tanimoto a ser usado na filtragem, que irá reduzir os dados por eliminação de objetos com um grau de semelhança maior que o coeficiente escolhido. De seguida é iniciada a conversão das moléculas para SMILES e feita a comparação entres elas, sendo as moléculas escolhidas de forma aleatória Figura 4.10.

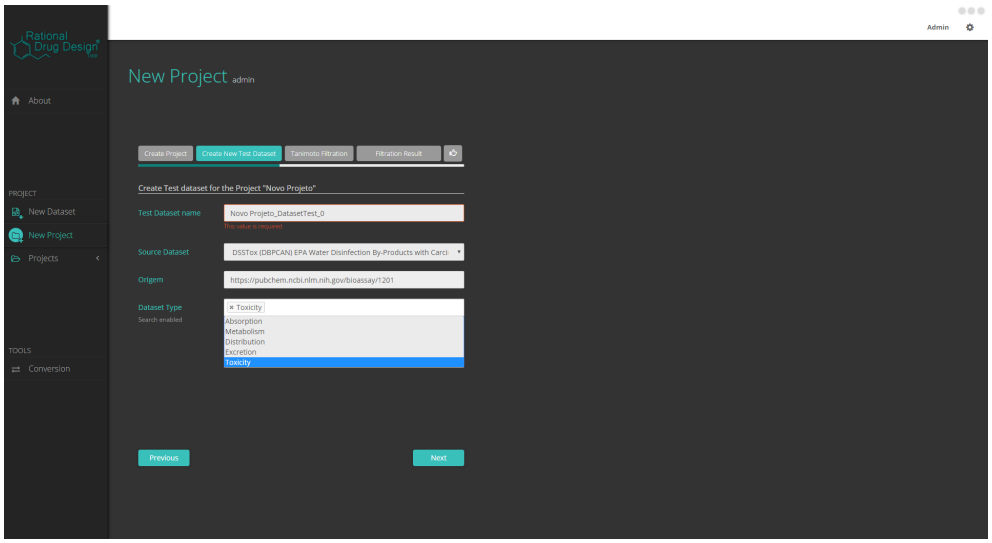

Figura 4.9: Identificação do tipo de dados a analisar.

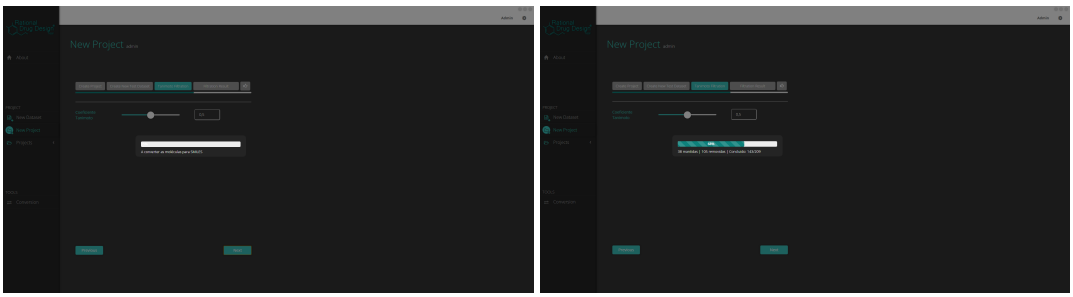

Figura 4.10: Conversão e filtragem das moléculas.

Como resultado da fase anterior de filtragem de moléculas, obtém-se os conjuntos de moléculas mantidas e de moléculas removidas. Esses conjuntos de moléculas são mostrados ao utilizador nesta fase Figura 4.11.

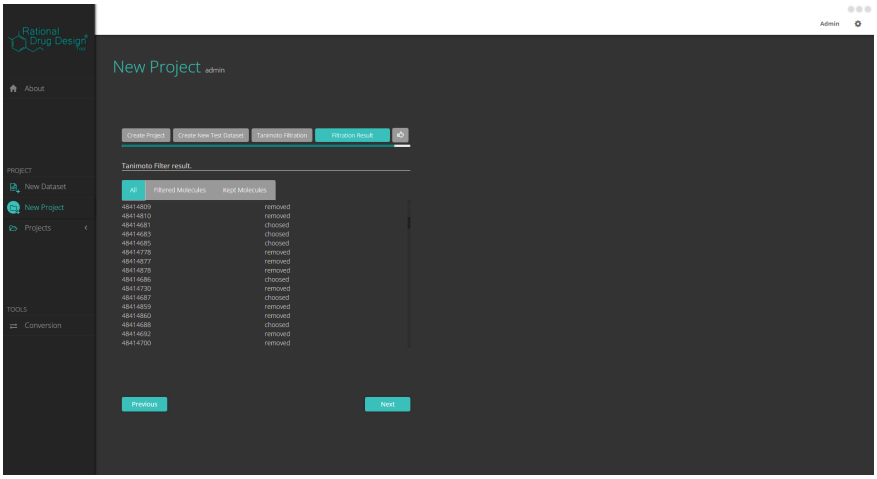

Figura 4.11: Resultado da aplicação do filtro.

Na ultima fase, é apresentado ao Utilizador a informação atual dos dados de teste que serão usados pelo projeto, tendo já o conjunto de dados original reduzido para uma amostra Figura 4.12, após a aplicação do algoritmo de Tanimoto .

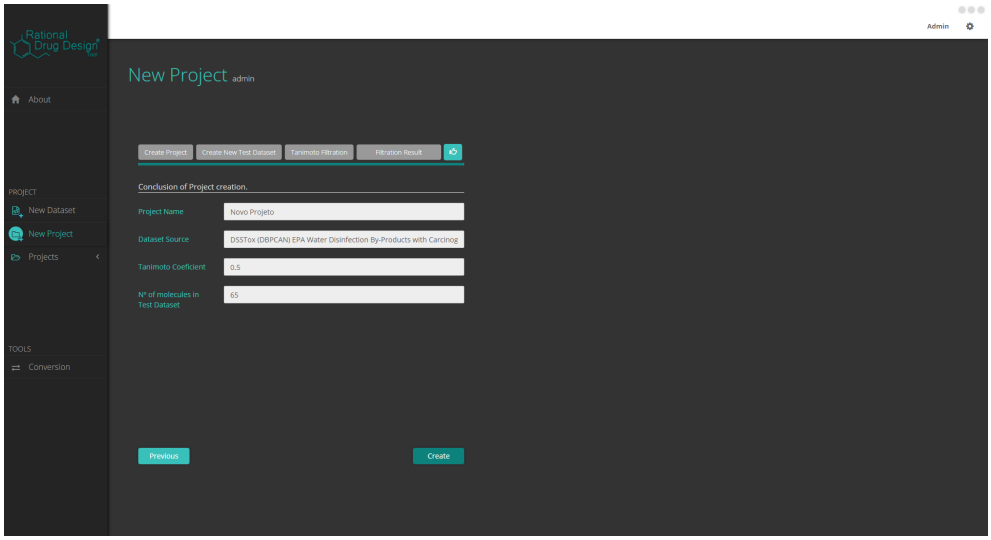

Figura 4.12: Finalizar a criação do projeto.

No final da criação de um novo projeto, será feita a transição para a vista de detalhes do projeto.

4.3.3 Projetos

Nesta vista é apresentada a lista de projetos do Utilizador, com uma informação breve sobre cada um deles. A partir da escolha de um projeto o Utilizador pode transitar para a vista deste Figura 4.13.

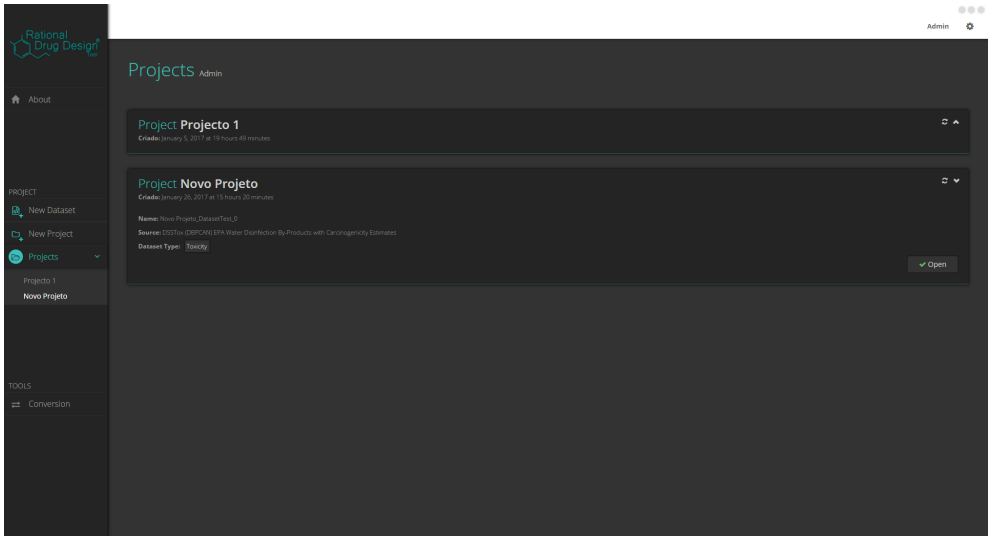

Figura 4.13: Lista de projetos do utilizador.

4.4 Projeto

A vista projeto é uma vista abstrata sobre a qual podem ser carregadas as três vistas seguintes: **detalhes**, **nova experiência WEKA** e **experiências WEKA**.

Tal como na vista abstrata anterior, o Utilizador pode alternar as sub-vistas deste grupo através da barra lateral e fechar o projeto na barra superior. Nesta vista o Utilizador tem, no fundo da barra lateral, informações sobre a amostra do projeto: numero de moléculas ativas e inativas, e o numero de moléculas usadas em comparação com o conjunto de dados inicial.

4.4.1 Detalhes

Nesta vista é apresentada a informação básica do projeto (nome, descrição, tipo de dados) e a informação do conjunto de dados inicial (descrição do ensaio realizado, protocolo seguido, tabela de moléculas e atividade, resultados e comentários) Figura 4.14.

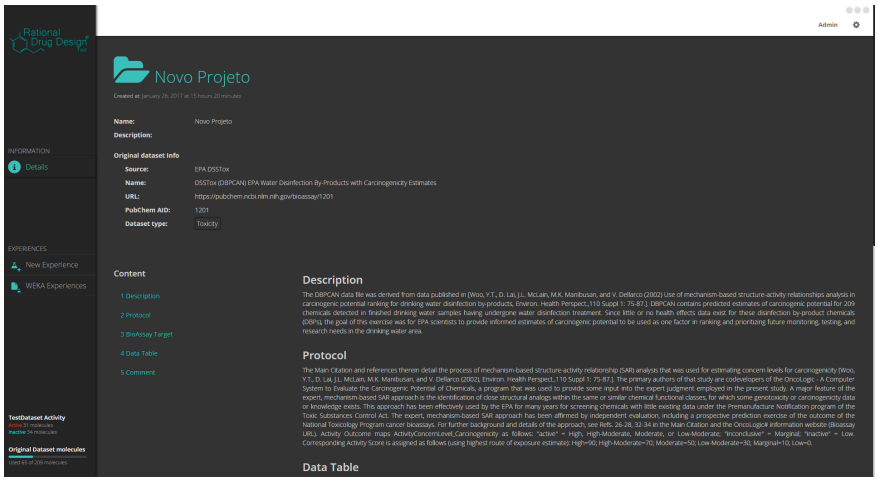

Figura 4.14: Detalhes de um projeto.

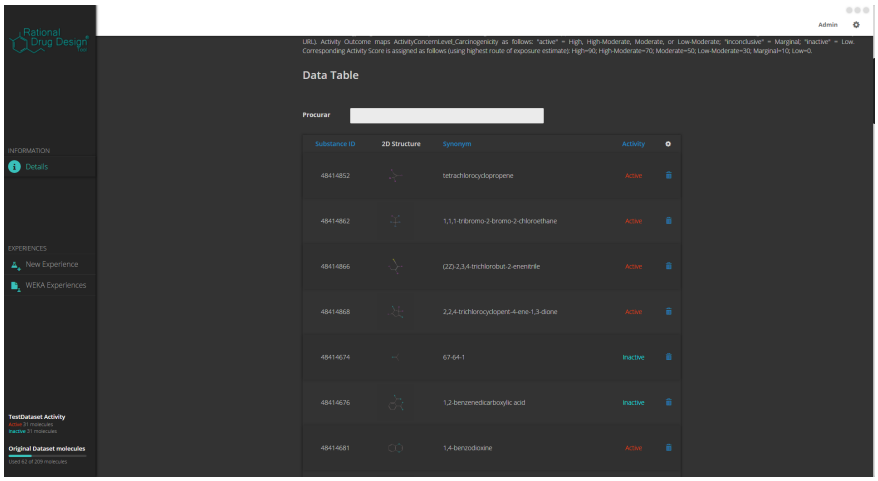

Figura 4.15: Tabela de moléculas da amostra do Projeto.

Na tabela de moléculas apresentada nesta vista, o Utilizador pode remover manualmente moléculas e visualizar a sua estrutura 2D Figura 4.15 e Figura 4.16.

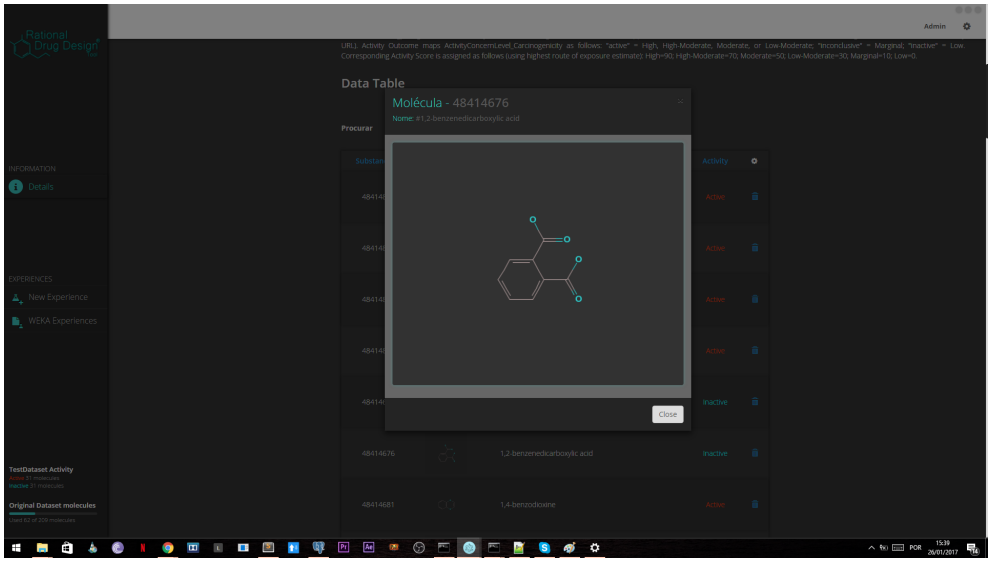

Figura 4.16: Visualização 2D de uma molécula.

4.4.2 Nova Experiência WEKA

Nesta vista é apresentado um formulário para a criação de uma nova experiência, sendo este dividido em duas fases. A transição entre as fazes é feita após a validação dos campos da corrente fase e da criação dos ficheiros ARFF a usar nos classificadores.

Na primeira fase, o Utilizador deve indicar o nome e a descrição a dar à experiência, seguida da escolha dos descritores a usar e do número máximo de valores em falta que um descritor pode ter Figura 4.17.

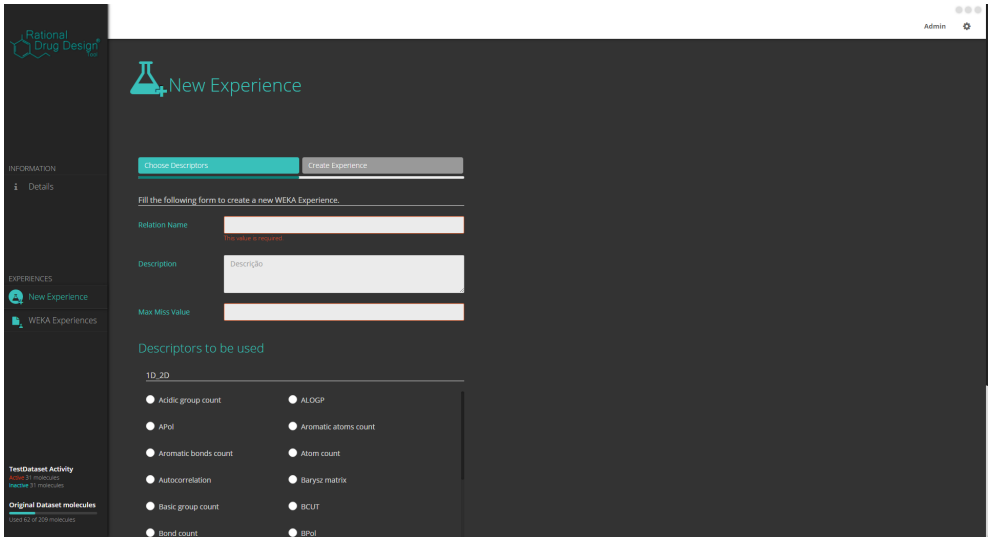

Figura 4.17: Formulário de criação de uma nova experiência.

Feito isto, é gerado um ficheiro ARFF com os dados a utilizar na experiência, sobre o qual é feita uma análise na fase seguinte.

Na segunda fase é apresentada ao Utilizador a informação relativa a cada descritor escolhido (percentagem de valores em falta, percentagem de valores únicos, entre outros) sob a forma de uma tabela e de um gráfico. Na Figura 4.18.

Figura 4.18: Análise dos descritores a usar na experiência.

Caso o Utilizador pretenda remover um descritor após a análise feita nesta fase, apenas tem que retroceder para a fase anterior e remover a seleção do descritor a retirar da experiência.

4.5 Experiência

A vista experiência é uma vista abstrata sobre a qual podem ser carregadas as três vistas seguintes: **modelos**, *novoModelo* e **modelo**.

A alternância entre as sub-vistas é feita na barra lateral, para alternar entre **modelos** e **novoModelo**, sendo a sub-vista **modelo** selecionada na sub-vista **modelos**. Na barra lateral o Utilizador tem como informação extra o numero de instâncias e de atributos a usar na experiência.

4.5.0.1 Modelos

Nesta vista é apresentada a informação básica da experiência, assim como as informações detalhadas dos atributos escolhidos e a lista de modelos criados, ilustrada na Figura 4.19 A partir desta vista o Utilizador pode abrir um **modelo** e testar um conjunto de moléculas desconhecidas Figura 4.20.

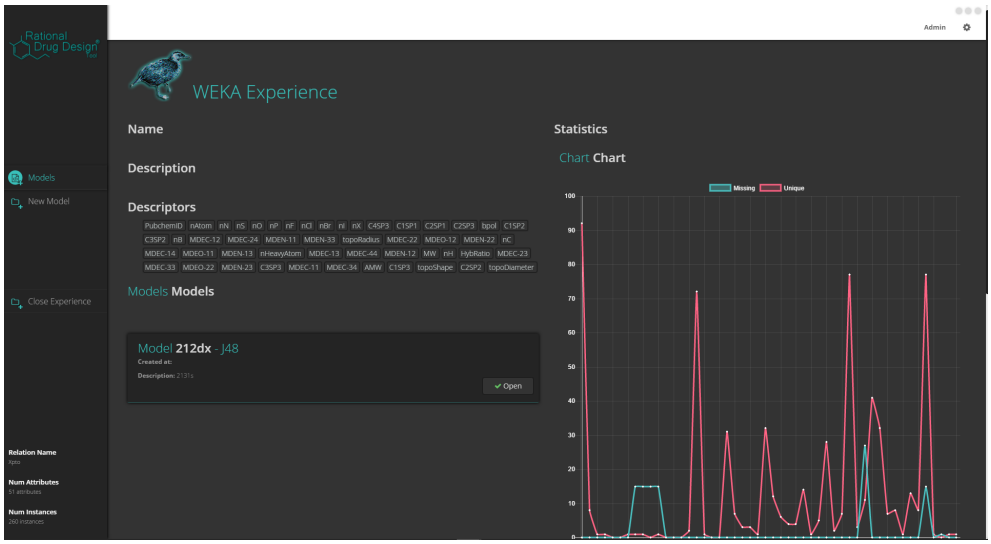

Figura 4.19: Informações da Experiência e lista de modelos.

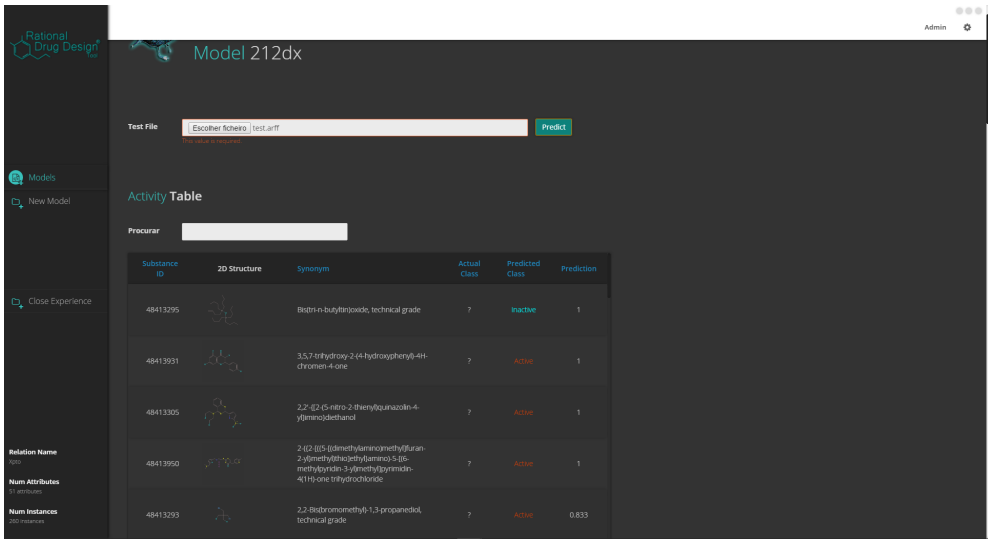

Figura 4.20: Efetuar a predição da atividade de moléculas.

4.5.0.2 Novo Modelo

A vista **novoModelo** é a mais complexa da aplicação, a nível de *output* de informação, processamento de dados e análise visual.

Nesta interface o utilizador pode usar diferentes técnicas para classificar o conjunto de dados da experiência, analisar os desempenhos das diferentes técnicas e escolher uma técnica para construir um modelo preditivo.

No formulário inicial, o Utilizador deve indicar o nome e a descrição do modelo a criar, ilustrado na Figura 4.21. De seguida o Utilizador pode escolher de entre 8 algoritmos de extração de conhecimento, qual pretende usar e quais as opções a fornecer ao algoritmo.

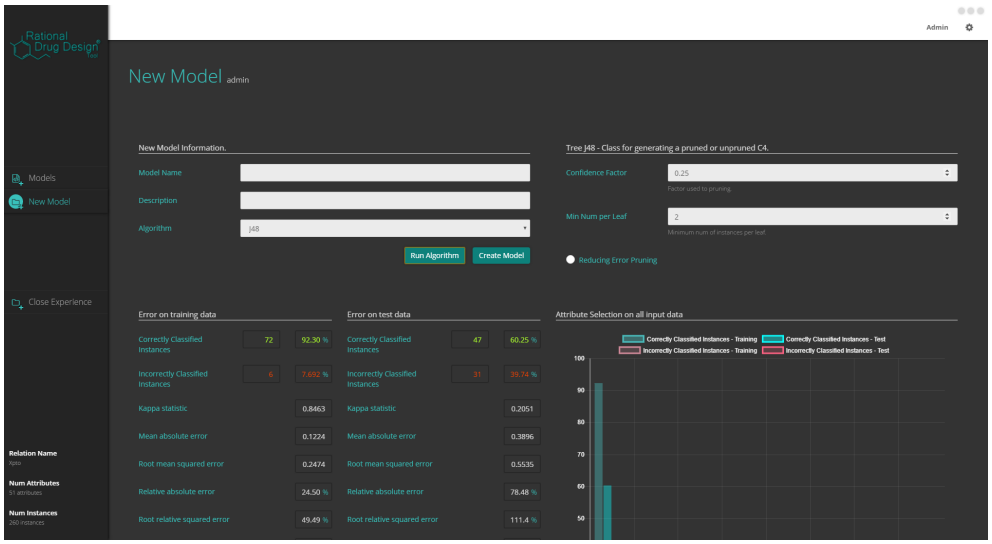

Figura 4.21: Escolher e executar um algoritmo de classificação.

Após fornecer as opções e correr o algoritmo o Utilizador pode analisar o desempenho da técnica através dos erros verificados nos dados de treino e nos dados de teste, como ilustrado no lado esquerdo da Figura 4.22. Os erros apresentados para os dados de treino e de teste, são referentes ao ultimo algoritmo corrido, enquanto no gráfico de barras, ilustrado no lado direito da Figura 4.22, é guardado, para cada algoritmo, as taxas em que o algoritmo teve o melhor desempenho nos dados de teste. O Utilizador pode ativar e desativar barras do gráfico.

Figura 4.22: Análise dos erros nos dados de treino e de teste.

A matriz de confusão para os dados da fase de treino e de teste, foi convertida numa linguagem mais intuitiva ao Utilizador, como ilustrado no fundo do lado direito da Figura 4.22.

Sobre os algoritmos, pode ser realizada outra análise, baseada na matriz de confusão, através da relação entre os Verdadeiros Positivos e os Falsos Positivos. Esta informação é apresentada num

espaço ROC, onde as instâncias de todos os algoritmos corridos são colocadas, como ilustrado no lado direito da Figura 4.23. Desta forma o Utilizador pode analisar instâncias de algoritmos diferentes para identificar o que obteve melhor desempenho até ao momento. É também fornecida ao utilizador a informação das medidas de performance, como ilustrado no lado esquerdo da Figura 4.23.

Figura 4.23: Métricas de desempenho e espaço ROC.

Sumário do Capitulo

Neste capitulo foram apresentadas as interfaces que compõem a aplicação, as transições existentes entre elas e as interações que o Utilizador pode realizar em cada uma delas.

Capítulo 5

Conclusões e Trabalho Futuro

5.1 Satisfação dos Objetivos

Para esta tese foi proposta a construção de uma aplicação que integrasse diferentes ferramentas úteis no desenvolvimento de novos fármacos, e que ao mesmo tempo fosse de fácil utilização.

Os objetivos propostos foram cumpridos, tendo sido desenvolvida uma aplicação capaz de carregar ficheiros no formato SDF, converter formatos de diversos ficheiros, calcular descritores moleculares, aplicar técnicas de amostragem e limpeza dos conjuntos de dados, realizar experiências com amostras através de algoritmos de aprendizagem, construir modelos preditivos, avaliar modelos preditivos e usar modelos em conjuntos de dados desconhecidos. Na fase de construção de modelos, por meio da análise dos desempenhos de diferentes algoritmos, foram implementados métodos de visualização do desempenho e de comparação entre eles.

Em suma, os objetivos estipulados foram atingidos, tendo sido construida uma aplicação capaz de auxiliar um especialista bioquímico no processo de desenho de um novo fármaco.

5.2 Trabalho Futuro

O possível trabalho futuro será a extensão das funcionalidades implementadas, através da integração de outros bancos de dados, como o *ChemSpider*, aperfeiçoar o calculo de descritores para a máquina onde a aplicação corre, adicionar mais métodos de amostragem e limpeza dos conjuntos de dados, adicionar algoritmos de *clustering* e regressão, melhorar o processo de avaliação de desempenho dos algoritmos, e melhorar a interface e interação do Utilizador com a aplicação.

Referências

[Die98] T. G. Dietterich. Approximate statistical tests for comparing supervised classification learning algorithms. *Neural Computation*, 10(7):1895–1924, 1998.

[DRO01] Stork D. G. Duda R. O., Hart P. E. *Pattern Classification*. Wiley-Interscience, 2001.

[eBJA03] Monard M. C. e Baranauskas J. A. Conceitos de aprendizando de máquina. *Sistemas inteligentes- Fundamentos e aplicações*, pages 89–114, 2003.

[EF16] e Ian H. Witten (2016) Eibe Frank, Mark A. Hall. *The WEKA Workbench*. Online Appendix for Data Mining: Practical Machine Learning Tools and Techniques, Fourth edition, 2016.

[Faw05] T. Fawcett. An introduction to ROC analysis. *Pattern Recognition Letters*, pages 861–874, 2005.

[MH] Geoffrey Holmes Bernhard Pfahringer Peter Reutemann e Ian H. Witten (2009) Mark Hall, Eibe Frank. *The WEKA Data Mining Software: An Update*, volume 11. SIGKDD Explorations.

[NMO11] C A James C Morley T Vandermeersch e G R Hutchison N M O'Boyle, M Banck. Open babel: An open chemical toolbox, 2011. [Última modificação Outubro 2011]. URL: http://openbabel.org, doi:10.1186/1758-2946-3-33.

[oW] University of Waikato. Attribute-relation file format. [Stable release: 3.8.1 (stable) / 14 Abril 2016]. URL: http://www.cs.waikato.ac.nz/ml/weka/arff.html, https://weka.wikispaces.com/ARFF.

[Weg01] Joerg Kurt Wegner. Joelib/joelib2, 2001. URL: https://sourceforge.net/projects/joelib/.

[Wik17a] Wikipedia. Chemical file format, 2017. [Online; Última modificação 29 Janeiro 2017]. URL: https://en.wikipedia.org/wiki/Chemical_file_format.

[Wik17b] Wikipedia. Chemical table file, 2017. [Online; Última modificação 2 Janeiro 2017]. URL: https://en.wikipedia.org/wiki/Chemical_table_file.

[Wik17c] Wikipedia. Simplified molecular-input line-entry system, 2017. [Online; Última modificação 20 Fevereiro 2017]. URL: https://en.wikipedia.org/wiki/Simplified_molecular-input_line-entry_system.

[Yap11] Chun Wei Yap. Padel-descriptor: An open source software to calculate molecular descriptors and fingerprints. *Journal of Computational Chemistry*, 32(7):1466–1474, 2011. URL: http://dx.doi.org/10.1002/jcc.21707, doi:10.1002/jcc.21707.